A flexible regression approach using GAMLSS in R.

Bob Rigby and Mikis Stasinopoulos

November 13, 2009

Preface

This book is designed for a short course in GAMLSS given at the University of Lancaster.

Contents

List of Figures

List of Tables

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 the GAMLSS framework. Section 2.2 defines the GAMLSS framework and contains a description of the type of regression models that the GAMLSS framework is capable of modelling. Section 2.3 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 are 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 $0, 1, 2, \ldots, \infty$. 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 the response variable takes values $0, 1, 2, \ldots, n$ 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. The important topic of centile estimation is given in Chapter 10.

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. $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ 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 the show notation that will be used throughout this book.

	Systematic part
Y:	univariate response variable
$\mathbf{y}$:	vector of observed values of the response variable, i.e. $(y_1, y_2, \ldots, y_n)^\top$
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
	Distributions and parameters
$f_Y()$:	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 $\boldsymbol{\pi} = (\pi_1, \pi_2 \ \ldots, \pi_k)^\top$
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
$\boldsymbol{i}()$:	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

2.1 Introduction

This Chapter serves as an introduction to generalized additive models for location, scale and shape (GAMLSS). This section gives a introduction to regression models. Section 2.2 describes the GAMLSS statistical framework. That is, the GAMLSS statistical model and its sub-models, the different distributions, the different additive terms and the different algorithms used within GAMLSS. Section 2.3 describes the different **gamlss** packages in R while Section 2.4 demonstrates of the `gamlss()` function.

The models we are dealing with in this book are models with *response* or *target* variable, (the y-variable) and possibility many *explanatory*, *input* or *independent* variables, (the x's).

2.1.1 Linear Model (LM)

A simple but effective model, (which served the statistical community well for the main part of the last century), is the linear regression model

$$Y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \epsilon_i \tag{2.1}$$

where Y_i for $i = 1, 1, \ldots, n$ are the response random variables and $(x_{1i}, \ldots, x_{pi})$ for $i = 1, \ldots, n$, are observed values from n observations in the data and p is the number of explanatory variables. The ϵ_i, for $i = 1, \ldots, n$, are the *errors* or *disturbances* and are assumed to be independently identically distributed random variables with zero means and a constant variance.

Model (2.1) can be written more conveniently in a matrix form as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.2}$$

where $\mathbf{Y}$ and $\boldsymbol{\epsilon}$ are $n \times 1$ vectors, $\mathbf{X}$ is a known $n \times p$ matrix and $\boldsymbol{\beta}$ is $p \times 1$ vector. The unknown quantities in (2.2) are the parameters $\boldsymbol{\beta}$ which can be estimated minimizing the sum of squares of the errors:

$$\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = \sum_{i=1}^{n} \epsilon_i^2. \tag{2.3}$$

Minimizing (2.3) with respect to $\boldsymbol{\beta}$ results in the least squares estimator for $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \tag{2.4}$$

While the least squares solution in (2.4) provides estimates for the coefficients $\boldsymbol{\beta}$, it does not provide a framework of testing the significant of those coefficients. This comes with the additional assumption that that the errors are not only independently identically distributed with zero means and constant variance, but also that they follow a normal distribution, i.e. $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix. Summarizing the model is now given by:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \text{where} \qquad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n). \tag{2.5}$$

By taking expectations with respect to $\boldsymbol{\epsilon}$ in equation (2.5) and by noting that any linear function of a normally distributed variable is normally distributed itself we can rewrite the model in (2.5) as:

$$\mathbf{y} \sim N(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n), \qquad \text{where} \qquad \boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}. \tag{2.6}$$

The reason why we prefer the formulation in (2.6) rather than in (2.5) is that it easier to generalize to non-normal distributions (which is the main subject of this book). Note also that in both formulations above the expectations are conditioned on the observed values of the explanatory variables, that is, we model the response given the $\mathbf{X}$'s.

The model (2.6) models the relationship between the mean of Y, $E(Y) = \mu$, and the x's linearly. We often refer to the assumed mathematical relationship between (any) parameter of the distribution of Y and x's as the *systematic* part of the statistical model. The assumption related to the variation of Y is referred to as the *stochastic* component of the statistical model.

The *likelihood* function of a model is the probability of observing the sample (given the parameters), so in the case of model (2.6) we have:

$$L\left(\boldsymbol{\beta}, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}} exp \left\{ -\frac{1}{2\sigma^2} \left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right)^\top \left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right) \right\} \tag{2.7}$$

with log-likelihood

$$\ell\left(\boldsymbol{\beta}, \sigma^2\right) = \frac{n}{2} log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right)^\top \left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right) \tag{2.8}$$

Note the maximising the log likelihood in (2.8) for $\boldsymbol{\beta}$ is equivalent of minimising the least squares quantity $\left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right)^\top \left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right)$ in (2.3). So in this case the maximum likelihood estimator (MLE) and Least Squares Estimator for $\boldsymbol{\beta}$ in (2.4) are identical. The MLE for σ^2 is given by

$$\hat{\sigma}^2 = \frac{\left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)^\top \left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)}{n}. \tag{2.9}$$

The MLE for σ^2, $\hat{\sigma}^2$ is a biased estimator so the unbiased version

$$s^2 = \frac{\left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)^\top \left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)}{n - p} \tag{2.10}$$

is often used instead, where p is the rank of the matrix $\mathbf{X}$. Sometimes the unbiased estimator in (2.10) is referred as the *REML* estimate of σ.

The corresponding *estimates* of $\boldsymbol{\beta}$ and σ^2 are given by substituting the observed values $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\top$ for the random variables $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)^\top$ giving for example the estimate $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$ for $\boldsymbol{\beta}$.

Sub-models of the linear model

An important point here is that the the linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ has a a lot of sub-models which in the statistical literature have special names. An explanatory factor A (which is a categorical variable) with p levels can be expressed as multiple explanatory variables (dummy or indicator variables for each level of factor) and hence included in a linear model. A *one way analysis of variance* (ANOVA) is a linear model where the matrix of explanatory variables $\mathbf{X}$ contains a single *factor* A only. When we have two explanatory factors A_1 and A_2 then we have a *two way analysis of variance.* A combination of explanatory factors and continuous explanatory variables is called *analysis of covariance* or ANACOVA.

Model specification

In practice there are several problems that could arise from the specification of the linear model:

- The relationship between the response variable and the explanatory variables may not be linear.
- The error terms and consequently the response variables may not be normally distributed.
- The error terms may not be independent of each other.
- The variance of the error term (and therefore the response variable) may not be is not constant over the observations.

The Generalized Linear Model discussed in the next section partially addresses the first and second problems.

2.1.2 Generalized Linear Model (GLM)

Equation (2.6) for the linear model allows generalization to the generalized linear models (GLM), Nelder and Weddeburn (1972). Firstly the normal distribution for Y_i is replaced by an exponential family distribution (denoted EF in general), and secondly a monotonic *link* function $g(.)$ relating μ_i the mean of Y_i, to the linear predictor η_i is introduced:

$$\begin{aligned} Y_i &\sim EF(\mu_i, \phi) \\ g(\mu_i) &= \eta_i = \mathbf{x}_i^\top \beta \end{aligned} \tag{2.11}$$

independently for $i = 1, 2, \dots, n$. In vector form this is represented as:

$$\begin{aligned} \mathbf{Y} &\sim EF(\boldsymbol{\mu}, \phi) \\ g(\boldsymbol{\mu}) &= \boldsymbol{\eta} = \mathbf{X}^\top \beta. \end{aligned} \tag{2.12}$$

The exponential family distribution $EF(\mu, \phi)$ is defined by the probability (density) function $f_Y(y; \mu, \phi)$ of Y having the form:

$$f_Y(y; \mu, \sigma) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y, \phi)\right\} \tag{2.13}$$

where $E(Y) = \mu = b^{'}(\theta)$ and $V(Y) = \phi V(\mu)$ where the *variance function* $V(\mu) = b^{''}[\theta(\mu)]$. The form of (2.13) includes many important distributions including the normal, Poisson, gamma,inverse Gaussian and Tweedie (Tweedie, 1984) distributions having variance functions $V(\mu) = 1, \mu, \mu^2, \mu^3$ and μ^p for $p < 0$ or $p > 1$, respectively, and also binomial and negative binomial distributions with variance functions $V(\mu) = \frac{\mu(1-\mu)}{N}$ and $V(\mu) = \mu + \frac{\mu}{\phi}$ respectively.

2.1.3 Generalized Additive Model (GAM

2.1.4 Generalized Linear Mixed model (GLMM)

2.2 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)).

2.2.1 The model

A GAMLSS model assumes that, for $i = 1, 2, \ldots, n$, 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 $i = 1, 2, \ldots, n$, 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 $\mathbf{Y}^\top = (Y_1, Y_2, \ldots, Y_n)$ 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 $g_k(.)$ be a known monotonic link function relating the distribution parameter $\boldsymbol{\theta}_k$ to predictor $\boldsymbol{\eta}_k$:

$$g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = \mathbf{X}_k\boldsymbol{\beta}_k + \sum_{j=1}^{J_k} \mathbf{Z}_{jk}\boldsymbol{\gamma}_{jk}, \tag{2.14}$$

i.e.

$$g_1(\boldsymbol{\mu}) = \boldsymbol{\eta}_1 = \mathbf{X}_1\boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} \mathbf{Z}_{j1}\boldsymbol{\gamma}_{j1}$$

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

where $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, $\boldsymbol{\nu}$ $\boldsymbol{\tau}$, and, for $k = 1, 2, 3, 4$, $\boldsymbol{\theta}_k$ and $\boldsymbol{\eta}_k$ are vectors of length n, $\boldsymbol{\beta}_k^{\top} = (\beta_{1k}, \beta_{2k}, \ldots, \beta_{J'_k k})$ 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 (2.14) 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 (2.14), then we have the *semi-parametric additive* formulation of GAMLSS given by

$$
g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = \mathbf{X}_k\boldsymbol{\beta}_k + \sum_{j=1}^{J_k} h_{jk}(\mathbf{x}_{jk}) \tag{2.15}
$$

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 $j = 1, 2, \ldots, J_k$ 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,

$$
g_1(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = \mathbf{X}_k\boldsymbol{\beta}_k. \tag{2.16}
$$

Model (2.15) 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):

$$
g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = h_k(\mathbf{X}_k, \boldsymbol{\beta}_k) + \sum_{j=1}^{J_k} h_{jk}(\mathbf{x}_{jk}) \tag{2.17}
$$

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 (2.17) 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 (2.17) is reduced to a *non-linear parametric* GAMLSS model:

$$g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = h_k(\mathbf{X}_k, \boldsymbol{\beta}_k). \tag{2.18}$$

If, in addition, $h_k(\mathbf{X}_k, \boldsymbol{\beta}_k) = \mathbf{X}_k^\top \boldsymbol{\beta}_k$ for $i = 1, 2, \ldots, n$ and $k = 1, 2, 3, 4$ then (2.18) reduces to the linear parametric model (2.16). Note that some of the terms in each $h_k(\mathbf{X_k}, \boldsymbol{\beta}_\mathbf{k})$ 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 (2.16) or (2.18) as a *parametric* GAMLSS model.

The parametric vectors $\boldsymbol{\beta}_k$ and the random effects parameters $\boldsymbol{\gamma}_{jk}$, for $j = 1, 2, \ldots, J_k$ 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 $\ell_p(\boldsymbol{\beta}, \gamma)$ given by

$$\ell_p(\boldsymbol{\beta}, \gamma) = \ell(\boldsymbol{\beta}, \gamma) - \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{J_k} \lambda_{jk} \boldsymbol{\gamma}_{jk}^\top \mathbf{G}_{jk} \boldsymbol{\gamma}_{jk} \tag{2.19}$$

where $\ell(\boldsymbol{\beta}, \gamma) = \sum_{i=1}^{n} \log f_Y(y_i|\boldsymbol{\theta}^i) = \sum_{i=1}^{n} \log f_Y(y_i|\mu_i, \sigma_i, \nu_i, \tau_i)$ is the log likelihood function of the distribution parameters given the data. Note we used $(\boldsymbol{\beta}, \gamma)$ as argument in the penalized log-likelihood to emphasize what is maximized here; $(\boldsymbol{\beta}, \gamma)$ represent all the $\boldsymbol{\beta}_k{}'s$ and the $\boldsymbol{\gamma}_{jk}$'s, for $j = 1, 2, \ldots, J_k$ and $k = 1, 2, 3, 4$. More details about the algorithms used to maximize the penalized log likelihood ℓ_p are given in Section 2.2.4. For parametric GAMLSS model (2.16) or (2.18), $\ell_p(\boldsymbol{\beta}, \gamma)$ reduces to $\ell(\boldsymbol{\beta})$, and the β_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 2.2.2 and 2.2.3 respectively. The R function to fit a GAMLSS model is `gamlss()` in the package **gamlss**.

2.2.2 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.

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 that one parameterization has been implemented in the **gamlss** packages. For example, two parameter Weibull distribution can be parameterized as $f_Y(y|\mu, \sigma) = \left(\sigma y^{\sigma-1}/\mu^\sigma\right) \exp\left\{-(y/\mu)^\sigma\right\}$, denoted as WEI, or as

Distributions	R Name	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$	$\boldsymbol{\nu}$	$\boldsymbol{\tau}$
beta	`BE()`	`logit`	`logit`	`-`	`-`
beta inflated (at 0)	`BEOI()`	`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)	`LNO()`	`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 GA	`ZAGA()`	`log`	`log`	`logit`	`-`
zero adjusted IG	`ZAIG()`	`log`	`log`	`logit`	`-`

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

Distributions	R Name	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$	$\boldsymbol{\nu}$
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)

$f_Y(y|\mu,\sigma) = \sigma\mu y^{\sigma-1} e^{-\mu y^{\sigma}}$, denoted as WEI2, or as $f_Y(y|\mu,\sigma) = (\sigma/\beta)\,(y/\beta)^{\sigma-1} \exp\left\{-\,(y/\beta)^{\sigma}\right\}$ denoted as WEI3, for $\beta = \mu/\left[\Gamma(1/\sigma)+1\right]$. 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

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i|\boldsymbol{\theta}) \tag{2.20}$$

changes to

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} [F(y_{2i}|\boldsymbol{\theta}) - F(y_{1i}|\boldsymbol{\theta})] \tag{2.21}$$

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

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

$$f_Y(y|\boldsymbol{\psi}) \quad = \quad \sum_{k=1}^{K} \pi_k f_k(y|\boldsymbol{\theta}_k) \tag{2.22}$$

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 $k = 1, 2, \ldots, K$. Also $\sum_{k=1}^{K} \pi_k = 1$ and $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\pi})$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k)$ and $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$. 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

Additive terms	R function names
boosting	`boost()`
cubic splines based	`cs()`, `scs()`, `vc()`
decision trees	`tr()`
fractional and power polynomials	`fp()`, `pp()`
free knot smoothing (break points)	`fk()`
`loess`	`lo()`
neural networks	`nn()`
non-linear fit	`nl()`
penalized Beta splines based	`pb()`, `ps()`, `cy()`, `tp()`, `pvc()`
random effects	`random()` `ra()`, `rc()`, `re()`
ridge regression	`ri()`, `ridge()`
Simon Wood's gam	`ga()`

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

(e.g. 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**.

2.2.3 Available additive terms in GAMLSS

Equation (2.14) allows modelling of 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 functions `cs()` and `scs()` are 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 (2.15) 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 $\lambda \int_{-\infty}^{\infty} \left[h''(t)\right]^2 dt$. 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()` and codescs() each distinct x-value is a knot. The two functions `cs()` and codescs() differ on the way they are implemented and should produce identical results.

Varying coefficients

The function `vc()` and `pvc()` are varying coefficients functions. 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 `vc()` function implementation x has to be continuous or a two level factor with levels 0 and 1. In the `pvc` function, which uses penalized B-splines, x can be a factor with more than two levels.

Penalized splines

The functions `pb()`, `ps()`, `cy()`, `tp()`, `pvc()` are all based on penalised B-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 $\boldsymbol{\theta} = \mathbf{Z}(\mathbf{x})\boldsymbol{\gamma}$ where $\boldsymbol{\theta}$ can be any distribution parameter in a GAMLSS model, $\mathbf{Z}(\mathbf{x})$ is $n \times q$ B-spline basis design matrix for the explanatory variable $\mathbf{x}$ defined at q-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 $\mathbf{D}\boldsymbol{\gamma} \sim N_{q-r}(\mathbf{0}, \lambda^{-1}\mathbf{I})$ [or equivalently by $\boldsymbol{\gamma} \sim N_q(\mathbf{0}, \lambda^{-1}\mathbf{K}^{-})$ 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 rth 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 the argument `inter` in `pb()` [and of course where to put them; `pb()` and its older version `ps()` use 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 older function `ps()` function in **gamlss**, which is based on an `S-PLUS` function of Marx (2003), takes three degrees of freedom nor a default value if neither the degrees of freedom or the smoothing parameter are set by the user. If in the newer function `pb()`, the user has not specified the degrees of freedom nor the smoothing parameter the `pb()` estimates them using one of several different local methods: i) Maximum Likelihood (ML), ii) Generalized Cross Validation (GCV) or iii) Generalized Akaike information criterion (GAIC), with ML as the default.

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 $b_0 + b_1 x^{p_1} + b_2 x^{p_2} + ... + b_k x^{p_k}$. For given powers $p_1, p_2, ..., p_k$, 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 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 $b_0 + b_1 x^{p_1} + b_2 x^{p_2}$ 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 terns 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 that `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.

2.2.4 The GAMLSS algorithms

There are two basic algorithms used for maximizing the penalized likelihood given in (2.19). 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 $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)$ for a four parameter distribution]. Note that we have dropped the subscripts here to simplify the notation. However for many population probability (density) functions, $f_Y(y|\boldsymbol{\theta})$, the parameters $\boldsymbol{\theta}$ 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 $\boldsymbol{\theta} = (\mu, \sigma)$ 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 two algorithms RS and CG are demonstrated in Figure 2.1 which shows the maximum likelihood parameters estimation based on a sample from a Weibull, WEI(μ, σ), distribution. The contours are equal deviance contours (equal to twice the log likelihood).

The object of the algorithms is to maximize the penalized likelihood function ℓ_p, given by (2.19), for fixed hyperparameters $\boldsymbol{\lambda}$. Appendix 2B show how this can be achieved. For fully parametric models, (2.16) or (2.18), 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 $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)$ rather than for the $\boldsymbol{\beta}$ parameters. The algorithms have generally been found to be stable and fast using very simple starting values (e.g. constants) for the $\boldsymbol{\theta}$ parameters. Default values can be changed by the user if necessary. The function `nlgamlss()` in the package **gamlss.nl** provides a third algorithm for fitting parametric linear or non-linear GAMLSS models in equation (2.16) or (2.18) respectively. However the algorithm needs starting values for all the $\boldsymbol{\beta}$ parameters, rather than $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)$, 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 parametric 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).

2.2.5 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 $\hat{r}_i = \Phi^{-1}(\hat{u}_i)$ where

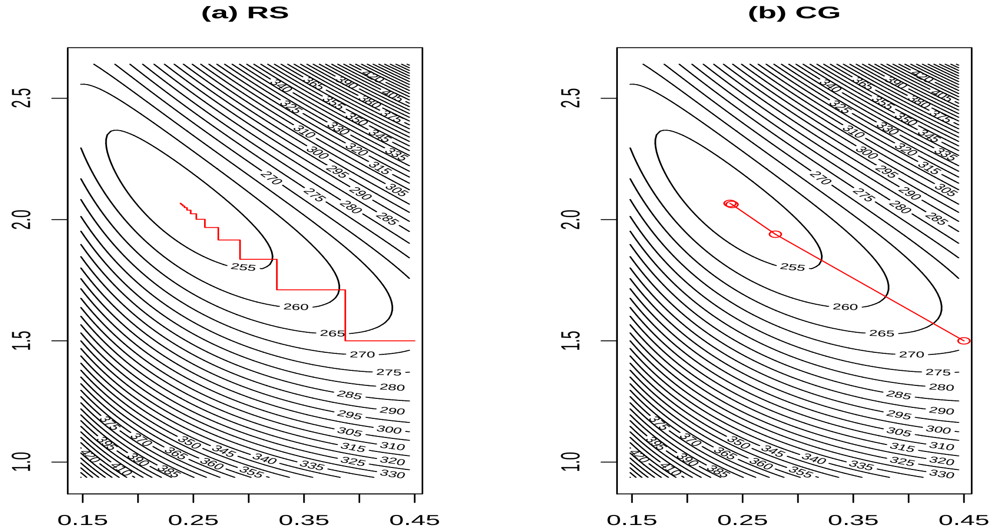

Figure 2.1: The two gamlss algorithms

Φ^{-1} is the inverse cumulative distribution function of a standard normal variate. The $\hat{u}_i$'s are defined differently for continuous and discrete response variables.

If y_i is an observation from a continuous response variable then $\hat{u}_i = F(y_i|\hat{\boldsymbol{\theta}}^i)$ where $u_i = F(y_i|\boldsymbol{\theta}^i)$ is the assumed cumulative distribution function for case i. The process is described diagrammatically in Figure 2.2. The top plot shows the probability distribution function for a specific observation y. The middle plot shows how, using the cumulative distribution function, the observation y is mapped into u. If the model is correctly specified u has a uniform distribution between zero and one. In the bottom figure u is transformed into a *z-score*, r, using $r = \Phi^{-1}(u)$, the inverse cumulative distribution function of a standard normal variate, so r will have a standard normal distribution. Note that $r_i = \Phi^{-1}\left[F(y_i|\boldsymbol{\theta}^i)\right]$. Similarly $\hat{u}_i$ is transformed to $\hat{r}$ by $\hat{r} = \Phi^{-1}(\hat{u}) = \Phi^{-1}\left[F(y|\hat{\boldsymbol{\theta}})\right]$ and $\hat{r}$ has an approximated standard normal distribution. So the normalized quantile residual r is the z-score corresponding to observation y based on its fitted distribution.

If y_i is an observation from a discrete integer response variable then $\hat{u}_i$ is a random value from the uniform distribution on the interval $[\hat{u}_1, \hat{u}_2] = \left[F(y_i - 1|\hat{\boldsymbol{\theta}}^i), F(y_i|\hat{\boldsymbol{\theta}}^i)\right]$ where $[u_1, u_2] = \left[F(y_i - 1|\boldsymbol{\theta}^i), F(y_i|\boldsymbol{\theta}^i)\right]$. The process is described in Figure 2.3. For a given probability distribution (top graph), the observed y value is transformed into an interval (u_1, u_2) (the shaded strip in middle plot). Then u is selected randomly from (u_1, u_2) and is transformed into the (randomized) z-score, r, (see the bottom graph). Hence, using the fitted cumulative distribution function, y is transformed to $\hat{u}$, randomly chosen from $(\hat{u}_1, \hat{u}_2)$, and then transformed to $\hat{r} = \Phi^{-1}(\hat{u})$.

Randomized residuals can be also used for interval or censored response variables. For example, for a right censored continuous response, $\hat{u}_i$ is defined as a random value from a uniform distribution on the interval $\left[F(y_i|\hat{\boldsymbol{\theta}}^i), 1\right]$.

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 exactly a standard normal distribution if the model is correct.

2.3 The gamlss packages

Section 2.3.1 provides some information on how to download the main **gamlss** packages. Section 2.3.2 shows the different functions available in the different packages available for modelling GAMLSS. Section 2.4 provides a basic introduction of the **gamlss** package.

2.3.1 How to input the GAMLSS framework packages

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

1. the original **gamlss** package for fitting a GAMLSS model
2. the **gamlss.add** package for extra additive teerms.
3. the **gamlss.boot** package for bootstrapping centiles.
4. the **gamlss.cens** package for fitting censored (left, right or interval) response variables.
5. the **gamlss.data** package for data used for demostration.
6. the **gamlss.dist** package for `gamlss.family` distributions
7. the **gamlss.mx** package for fitting finite mixture distributions.
8. the **gamlss.nl** package for fitting non-linear models
9. the **gamlss.tr** package for fitting truncated distributions.

The R and 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. First install R from CRAN. 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**.

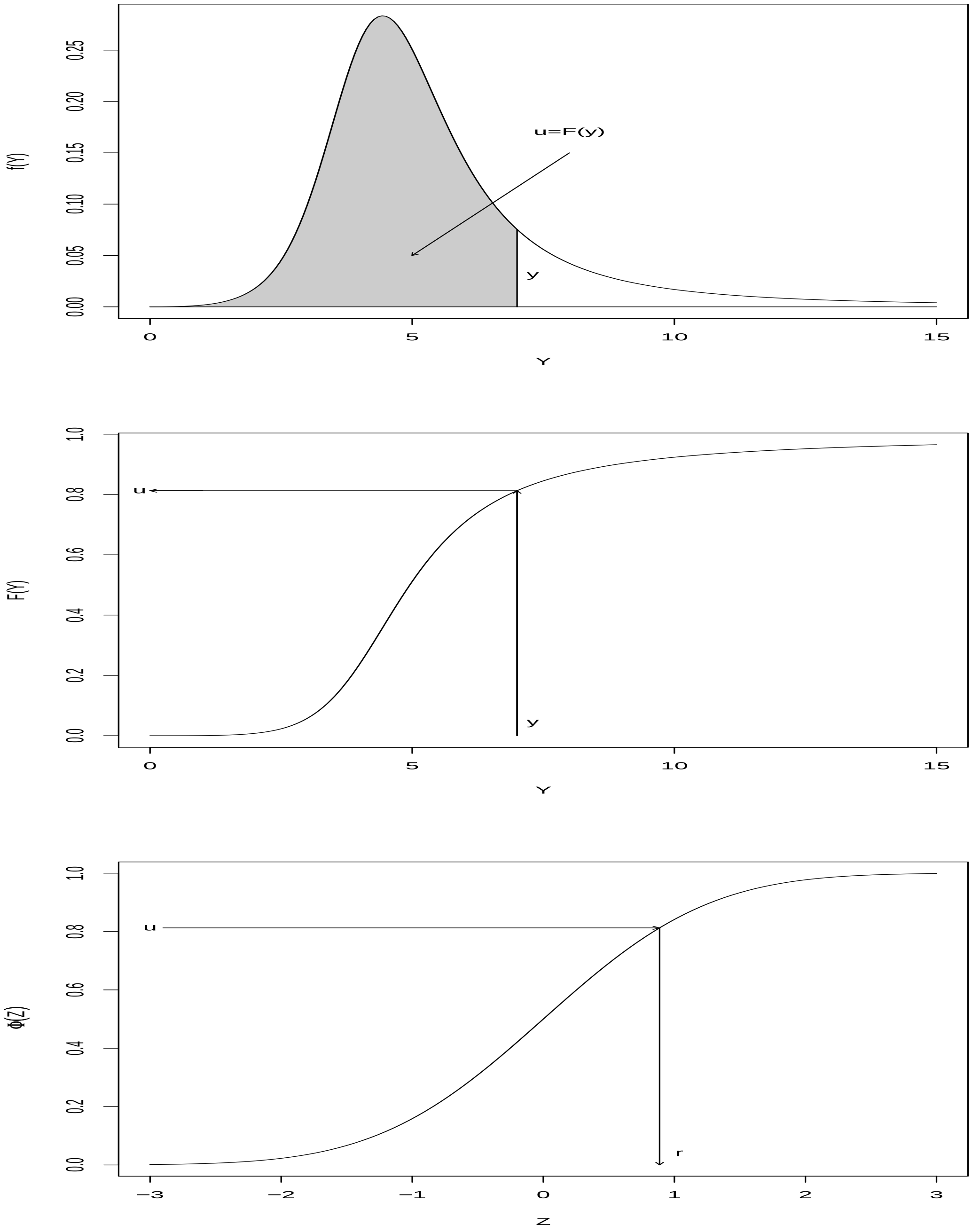

Figure 2.2: A description of how a (normalized quantile) residual r is obtained for continuous distributions. The functions plotted are the model probability density function $f(y)$, the cumulative distribution function $F(y)$ and inverse cumulative distribution function of a standard normal random variable $\Phi(z)$, using which y is transformed to u and then from u to r. The residual r is the z-score for the specific observation.

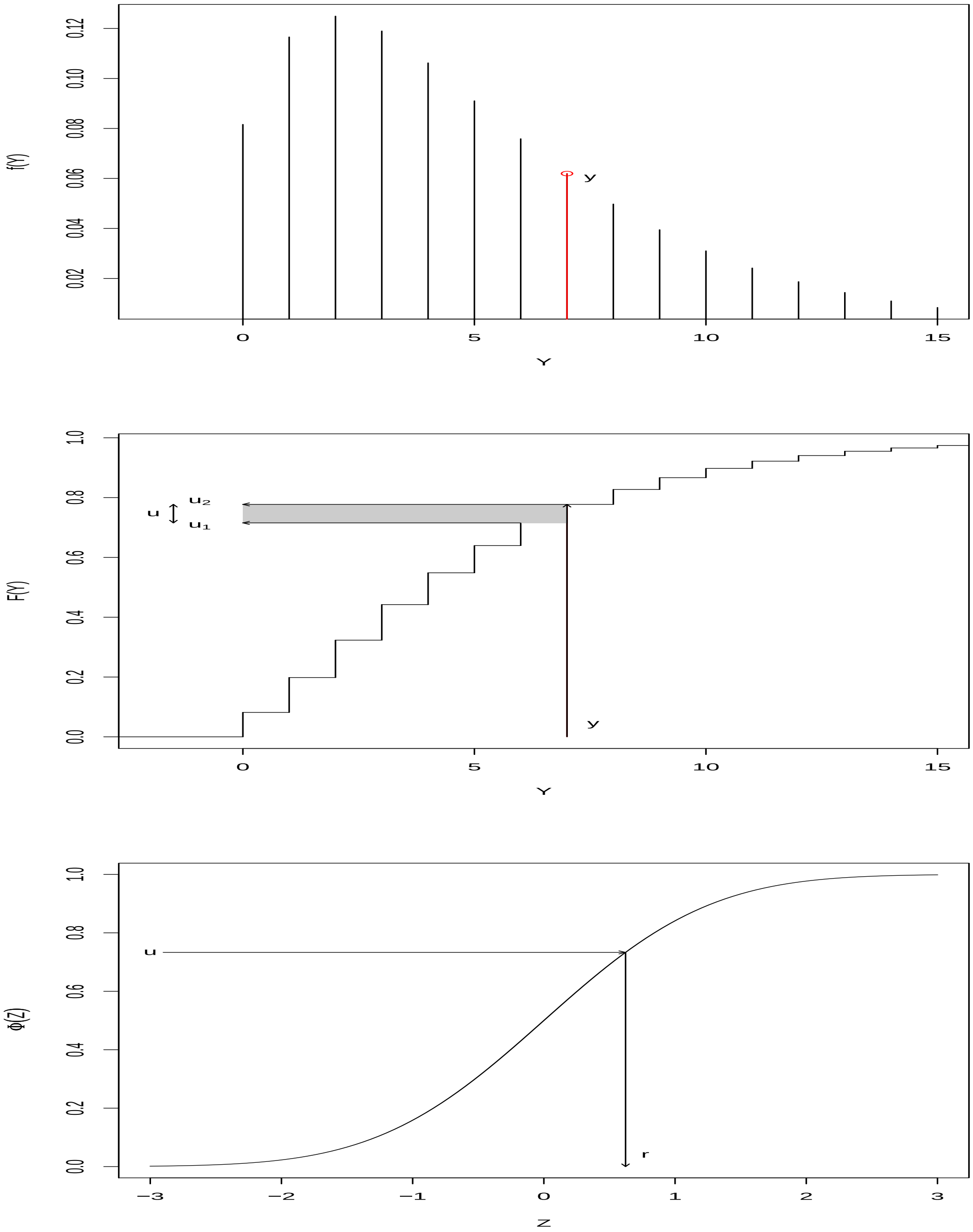

Figure 2.3: A description of how a (normalized randomized quantile) residual r is obtained for discrete distributions. The observed y is transformed to u, a random number between u_1 and u_2, then u is transformed to r. The residual r is the z-score for the specific observation.

2.3.2 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 2.4 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`)
- `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) or Section 2.2.5 for a definition of the normalized (randomized) quantile residuals.
- `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) terms 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 the 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 2.2.5.

- `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($\sharp$) information profile of a hyperparameter for a given penalty $\sharp$ 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.

2.4 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. The function `gamlss()` also can be used to fit models which can be fitted using the functions `glm()` of R and `gam()` of the recommended package **mgcv**. For parametric models `gamlss` and `glm()` should give identical results as far as the fitted values and the fitted coefficients for the mean are concern (given that the same distribution from the exponential family is fitted). For smoothing models `gamlss` results should be identical to the `gam()` results of package **gam** if the **gamlss** additive function `cs()` is used and for fixed degrees of freedom. For smoothing models where the additive **gamlss** function `pb()` is used, `gamlss()` and `gam()` of package **mgcv** should produce very similar results.

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.data** of dimensions 610 × 2

variables

`y` : abdominal circumference

`x` : gestational age

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

The 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 2.4. 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")
```

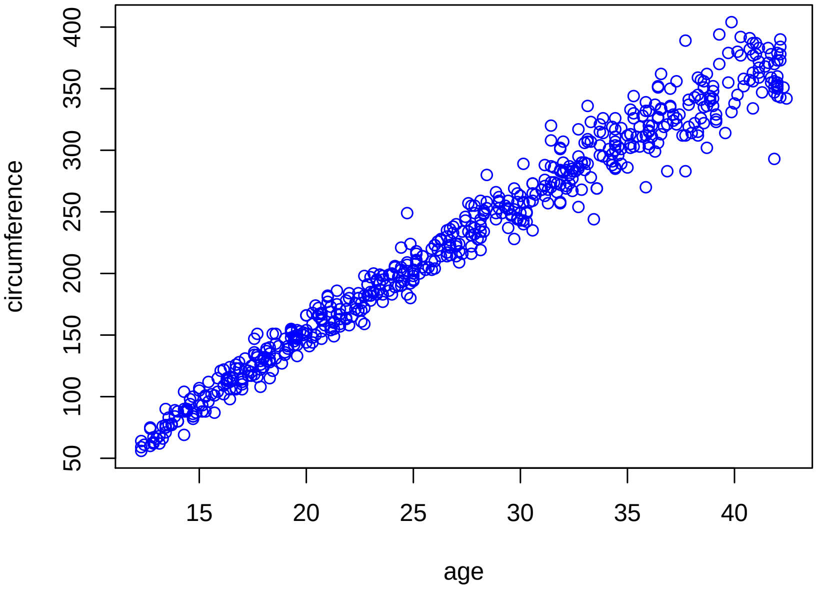

Figure 2.4: A plot of the abdominal circumference data

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

Fitting method: RS()

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

and then use them within the `gamlss()` function 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 parameters 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. β_{01}, β_{11}, β_{21}, β_{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 distribution 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 (2.16). 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 mu 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 is six, i.e. 5 for mu the mean, and 1 for the constant scale parameter sigma the standard deviation of the fitted normal distribution model.

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 mu 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 distribution parameters for both old and new data values of the explanatory variables.

To model both the mean, mu, and the scale parameter, sigma, 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 fitted (normalized randomized quantile) residuals of a 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 2.2.5.] 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
****************************************************************
```

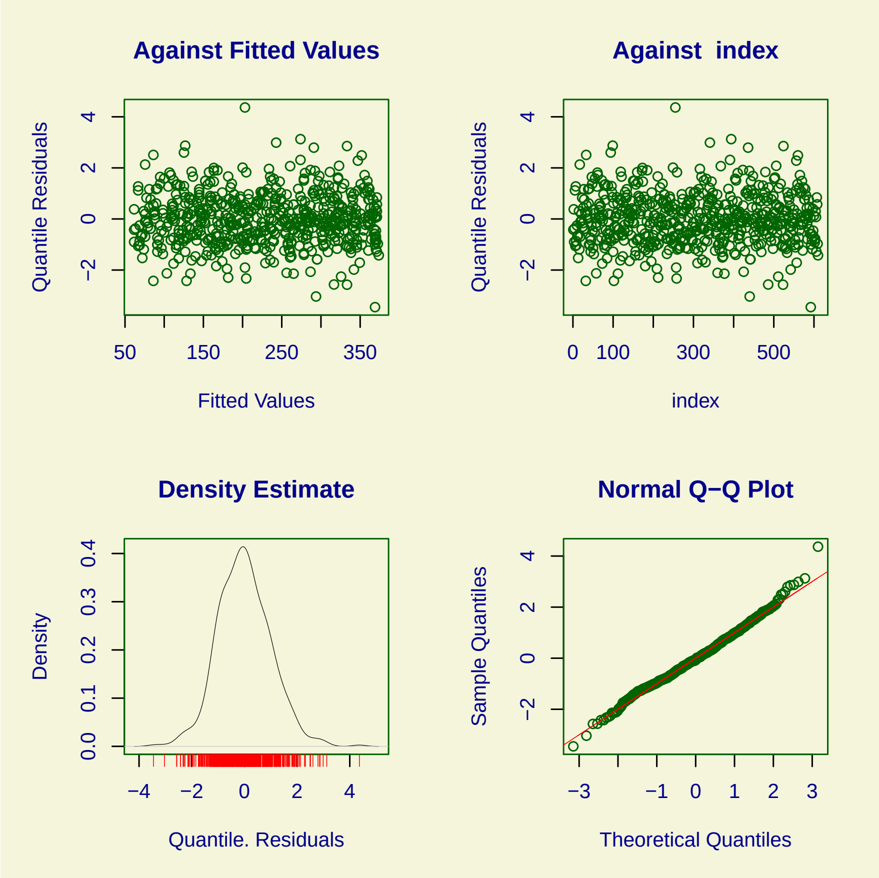

Figure 2.5: Residual plot from the fitted normal model abd2 with $\mu = cs(x, 3)$ and $\sigma = \; cs(x, 3)$

See Figure 2.5 for the plot. Figure 2.5 shows plots of the (normalized quantile) residuals: i) against the fitted values ii) against a index iii) a non-parametric kernel density estimate iv) a normal Q-Q 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 2.6(a) for the plot.

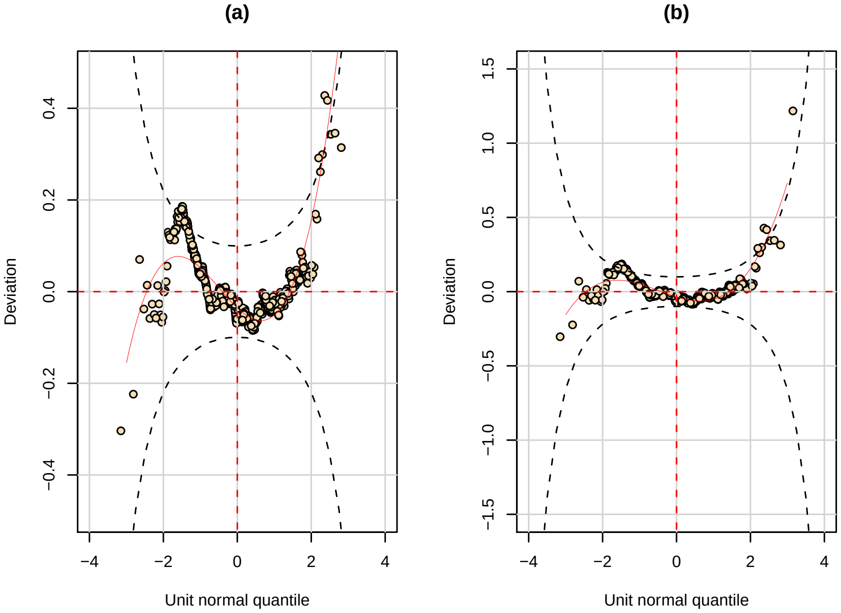

Figure 2.6: Worm plot from the normal fitted model abd2 with $\mu = cs(x, 3)$ and $\log(\sigma) = cs(x, 3)$, (a) with default deviation range, (b) with deviation range $(-1.5, 1.5)$.

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

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

Since there is no warning message all points have been included in the worm plot. See Figure 2.6(b) for the plot. [Clearly one point was omitted from Figure 2.6(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.

In model abd2 we fitted a smoothing function for both the μ and σ parameter by fixing extra the degrees of freedom for smoothing to be equal to three. This will gives 5 degrees for freedom for both μ and σ. The function pb() allows the smoothing parameters (and therefore the degrees of freedoms) to be estimated automatically within the GAMLSS algorithm.

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

GAMLSS-RS iteration 1: Global Deviance = 4786.697
GAMLSS-RS iteration 2: Global Deviance = 4785.695
GAMLSS-RS iteration 3: Global Deviance = 4785.696

> abd3$mu.df

[1] 5.679297

> abd3$sigma.df

[1] 2.002543
```

The estimated total degrees of freedom for smoothing are 5.679 and 2.0025 for μ and σ respectively. The locally estimated degrees of freedom for μ are a bit higher that fixed degrees of freedom used for models abd1 and abd2. The σ degrees of freedom are almost 2 indicating that we only need a linear model for x, that is the model with sigma.formula = ∼x.

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

```
> abd4 <- gamlss(y ~ lo(x, span = 0.4), sigma.formula = ~lo(x,
+     span = 0.4), data = abdom, family = NO)

GAMLSS-RS iteration 1: Global Deviance = 4785.719
GAMLSS-RS iteration 2: Global Deviance = 4785.286
GAMLSS-RS iteration 3: Global Deviance = 4785.28
GAMLSS-RS iteration 4: Global Deviance = 4785.279
```

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:

```
> abd5 <- gamlss(y ~ pb(x), sigma.formula = ~pb(x), data = abdom,
+     family = TF)

GAMLSS-RS iteration 1: Global Deviance = 4780.234
GAMLSS-RS iteration 2: Global Deviance = 4777.493
GAMLSS-RS iteration 3: Global Deviance = 4777.518
GAMLSS-RS iteration 4: Global Deviance = 4777.519
```

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 fitted log-likelihood function and $\sharp$ is a required penalty, e.g. $\sharp = 2$ for the usual Akaike information criterion or $\sharp = log(n)$ for the Schwartz Bayesian criterion. The function `deviance()` provides the global deviance of the model. Note that the GAMLSS global deviance is different from the deviance that 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 but depend in scale parameter) 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, abd4, abd5)

            df      AIC
abd5  8.789771 4795.099
abd3  7.681840 4801.060
abd2  9.999872 4804.718
abd4 10.667506 4806.614
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 GAIC($\sharp = 2$)] selects model `abd5` 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, abd4, abd5, k = 3)

            df      AIC
abd5  8.789771 4803.889
abd3  7.681840 4808.742
abd2  9.999872 4814.718
abd4 10.667506 4817.282
abd1  6.000680 4955.162
```

Hence, GAIC($\sharp = 3$) also selects model `abd5` as the best model.

2.5 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 (2.19) over the parameters $\boldsymbol{\beta}_k$ and terms $\boldsymbol{\gamma}_{jk}$ for $j = 1, 2, \ldots, J_k$ and $k = 1, 2, .., p$ 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 $k = 1, 2, \ldots, p$ and $s = 1, 2, \ldots, p$, 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 $i = 1, 2, ..., n$, 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 $\boldsymbol{\gamma}_{jk}^{(r,i,m)}$ stand for the current value of the vector $\boldsymbol{\gamma}_{jk}$ in the r^{th} outer, i^{th} inner and m^{th} backfitting cycle iteration and let $\boldsymbol{\gamma}_{jk}^{(r,i,.)}$ stand for the value of $\boldsymbol{\gamma}_{jk}$ 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 $\boldsymbol{\gamma}_{jk}^{(r,i+1,1)}$ for the $(i+1)^{th}$ inner cycle of the r^{th} outer cycle, for $j = 1, 2, \ldots, J_k$ and $k = 1, \ldots, p$. Note also, for example, $\boldsymbol{\gamma}_{jk}^{(r,i,c)}$ means the current (i.e. most recently) updated estimate of $\boldsymbol{\gamma}_{jk}$ 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 $\boldsymbol{\beta}_k$ and $\boldsymbol{\gamma}_{jk}$, for $j = 1, \ldots, J_k$ in the model successively for each θ_k in turn, for $k = 1, \ldots, p$. 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 $\boldsymbol{\theta}_k$, while in CG all weight matrices $\mathbf{W}_{ks}$ for $k = 1, 2, \ldots, p$ and $s = 1, 2, \ldots, p$ are evaluated *after* fitting *all* $\boldsymbol{\theta}_k$ for $k = 1, 2, \ldots, p$.

The RS algorithm is as follows :

- **Start**: Initialize fitted values $\boldsymbol{\theta}_k^{(1,1)}$ and random effects $\boldsymbol{\gamma}_{jk}^{(1,1,1)}$, for $j = 1, \ldots, J_k$ and $k = 1, 2, \ldots, p$. Evaluate the initial linear predictors $\boldsymbol{\eta}_k^{(1,1)} = g_k\left[\boldsymbol{\theta}_k^{(1,1)}\right]$, for $k = 1, 2, \ldots, p$.
- **START OUTER CYCLE** $r = 1, 2, \ldots$ **UNTIL CONVERGENCE. FOR** $k = 1, 2, \ldots, p$
    - **START INNER CYCLE** $i = 1, 2, \ldots$ **UNTIL CONVERGENCE** .
        - ∗ Evaluate the current $\mathbf{u}_k^{(r,i)}$, $\mathbf{W}_{kk}^{(r,i)}$ and $\mathbf{z}_k^{(r,i)}$
        - ∗ **START BACKFITTING CYCLE** $m = 1, 2, \ldots$ **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 $\boldsymbol{\beta}_k^{(r,i,m+1)}$.
        - ∗ For $j = 1, 2, \ldots, J_k$ smooth the partial residuals

$\varepsilon_{jk}^{(r,i,m)} = \mathbf{z}_k^{(r,i)} - \mathbf{X}_k\boldsymbol{\beta}_k^{(r,i,m+1)} - \sum_{t=1,t\neq j}^{J_k} \mathbf{Z}_{tk}\boldsymbol{\gamma}_{tk}^{(r,i,c)}$ using the shrinking (smoothing) matrix S_{jk} given by (**??**) to obtain the updated (and current) additive predictor term $\mathbf{Z}_{jk}\boldsymbol{\gamma}_{jk}^{(r,i,m+1)}$

    - **END BACKFITTING CYCLE**, on convergence of $\beta_k^{(r,i,.)}$and $\mathbf{Z}_{jk}\boldsymbol{\gamma}_{jk}^{(r,i,.)}$ and set $\boldsymbol{\beta}_k^{(r,i+1)} = \boldsymbol{\beta}_k^{(r,i,.)}$ and $\boldsymbol{\gamma}_{jk}^{(r,i+1)} = \boldsymbol{\gamma}_{jk}^{(r,i,.)}$ for $j = 1,2,\ldots,J_k$ and otherwise update m and continue the backfitting cycle.
    - **Calculate updated** $\boldsymbol{\eta}_k^{(r,i+1)}$ and $\boldsymbol{\theta}_k^{(r,i+1)}$.

  - **END INNER CYCLE** on convergence of $\boldsymbol{\beta}_k^{(r,.)}$ and the additive predictor terms $\mathbf{Z}_{jk}\boldsymbol{\gamma}_{jk}^{(r,.)}$ and set $\boldsymbol{\beta}_k^{(r+1,1)} = \boldsymbol{\beta}_k^{(r,.)}$, $\boldsymbol{\gamma}_{jk}^{(r+1,1)} = \boldsymbol{\gamma}_{jk}^{(r,.)}$, for $j = 1,2,\ldots,J_k$, $\boldsymbol{\eta}_k^{(r+1,1)} = \boldsymbol{\eta}_k^{(r,.)}$ and $\boldsymbol{\theta}_k^{(r+1,1)} = \boldsymbol{\theta}_k^{(r,.)}$, 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 : $\boldsymbol{\theta}_k^{(1,1)}$ and $\boldsymbol{\gamma}_{jk}^{(1,1,1)}$ for $j = 1,2,\ldots,J_k$ and $k = 1,2,\ldots,p$. Evaluate $\boldsymbol{\eta}_k^{(1)} = \boldsymbol{\eta}_k^{(1,1)} = g_k\left[\boldsymbol{\theta}_k^{(1,1)}\right]$ for $k = 1,2,\ldots,p$.
- **START OUTER CYCLE** $r = 1,2,\ldots$ **UNTIL CONVERGENCE**
- Evaluate and **fix** the current $\mathbf{u}_k^{(r)}$, $\mathbf{W}_{ks}^{(r)}$ and $\mathbf{z}_k^{(r)}$ for $k = 1,2,\ldots,p$ and $s = 1,2,\ldots,p$. Perform a single r^{th} step of the Newton-Raphson algorithm by:
  - **START INNER CYCLE** $i = 1,2,\ldots$ **UNTIL CONVERGENCE** :
    - **FOR** $k = 1,2,\ldots,p$
    - **START BACKFITTING CYCLE** $m = 1,2,\ldots$ **UNTIL CONVERGENCE**

$$X_k\boldsymbol{\beta}_k^{(r,i,m+1)} = \mathbf{H}_k^{(r)}\boldsymbol{\varepsilon}_{ok}^{(r,i,m)}$$

and for $j = 1,2,\ldots,J_k$

$$\mathbf{Z}_{jk}\boldsymbol{\gamma}_{jk}^{(r,i,m+1)} = \mathbf{S}_{jk}^{(r)}\boldsymbol{\varepsilon}_{jk}^{(r,i,m)}$$

    - **END BACKFITTING CYCLE**, on convergence of $\beta_k^{(r,i,.)}$ and $z_{jk}\boldsymbol{\gamma}_{jk}^{(r,i,.)}$ and set $\boldsymbol{\beta}_k^{(r,i+1)} = \boldsymbol{\beta}_k^{(r,i,.)}$ and $\boldsymbol{\gamma}_{jk}^{(r,i+1)} = \boldsymbol{\gamma}_{jk}^{(r,i,.)}$ for $j = 1,2,\ldots,J_k$ 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 $\boldsymbol{\beta}_k^{(r,.)}$ and the additive predictor terms $\mathbf{Z}_{jk}\boldsymbol{\gamma}_{jk}^{(r,.)}$ and set $\boldsymbol{\beta}_k^{(r+1,1)} = \boldsymbol{\beta}_k^{(r,.)}$, $\boldsymbol{\gamma}_{jk}^{(r+1,1)} = \boldsymbol{\gamma}_{jk}^{(r,.)}$, $\boldsymbol{\eta}_k^{(r+1)} = \boldsymbol{\eta}_k^{(r+1,1)} = \boldsymbol{\eta}_k^{(r,.)}$ and $\boldsymbol{\theta}_k^{(r+1,1)} = \boldsymbol{\theta}_k^{(r,.)}$, for $j = 1, 2, \ldots, J_k$ and $k = 1, 2, \ldots p$, 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 $j = 1, 2, \ldots, J_k$ and $k = 1, 2, \ldots, p$

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

$$\boldsymbol{\varepsilon}_{0k}^{(r,i,m)} = \mathbf{z}_k^{(r)} - \sum_{t=1}^{J_k} \mathbf{Z}_{tk}\boldsymbol{\gamma}_{tk}^{(r,i,c)} - {\mathbf{W}_{kk}^{(r)}}^{-1} \sum_{s=1,s\neq k}^{p} \mathbf{W}_{ks}^{(r)} \left[\boldsymbol{\eta}_s^{(r,c)} - \boldsymbol{\eta}_s^{(r)}\right]$$

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

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 $\boldsymbol{\eta}_k^{(r+1)}$ for $k = 1, 2, \ldots, p$, 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 $k = 1, 2, \ldots, p$ and $s = 1, 2, \ldots, p$, using the $\boldsymbol{\eta}_k^{(r+1)}(\alpha)$ for $k = 1, 2, \ldots, p$. 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 (2.19) over the parameters $\boldsymbol{\beta}_k$ and terms $\boldsymbol{\gamma}_{jk}$ for $j = 1, 2, \ldots, J_k$ and $k = 1, 2, .., p$ 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 (2.19) are obtained to give a Newton-Raphson step for maximizing (2.19) with respect to $\boldsymbol{\beta}_k$ and $\boldsymbol{\gamma}_{jk}$ for $j = 1, 2, \ldots, J_k$ and $k = 1, 2, ..., p$.

- (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 (2.19), using a Newton-Raphson algorithm. The first derivative (score function) and the second derivatives of ℓ_p with respect to $\boldsymbol{\beta}_k$ and $\boldsymbol{\gamma}_{jk}$ for all $j = 1, 2, \ldots, J_k$ and $k = 1, 2, .., p$ are evaluated at iteration r at the current predictors $\boldsymbol{\eta}_k^{(r)}$ for $k = 1, 2, ..., p$.

Let $\boldsymbol{\alpha}_k^T = \left[\boldsymbol{\beta}_k^T, \boldsymbol{\gamma}_{1k}^T, \boldsymbol{\gamma}_{2k}^T, \ldots, \boldsymbol{\gamma}_{J_k k}^T\right]$, $\mathbf{a}_k = \frac{\partial \ell_p}{\partial \boldsymbol{\alpha}_k}$ and $\mathbf{A}_{ks} = -\frac{\partial^2 \ell_p}{\partial \boldsymbol{\alpha}_k \partial \boldsymbol{\alpha}_s^T}$ for $k = 1, 2, .., p$ and $s = 1, 2, \ldots, p$, and let $\boldsymbol{\alpha}^T = [\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T, \ldots, \boldsymbol{\alpha}_p^T]$, $\mathbf{a} = \frac{\partial \ell_p}{\partial \boldsymbol{\alpha}}$ and $\mathbf{A} = -\frac{\partial^2 \ell_p}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^T}$.

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

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

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 $i = 1, 2, .., n$, for $k = 1, 2, .., p$ and $s = 1, 2, \ldots, p$ (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

$$\mathbf{Z}_{jk} \boldsymbol{\gamma}_{jk}^{(r+1)} = \mathbf{S}_{jk}^{(r)} \boldsymbol{\varepsilon}_{jk}^{(r)} \tag{2.23}$$

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

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

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 (2.23) is to apply weighted least squares estimation to an augmented data model given by:

$$\begin{bmatrix} \boldsymbol{\varepsilon}_{jk}^{(r)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{jk} \\ -\mathbf{D}_{jk} \end{bmatrix} \boldsymbol{\gamma}_{jk} + \begin{bmatrix} \mathbf{e}_{0k} \\ \mathbf{e}_{jk} \end{bmatrix} \tag{2.24}$$

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

$$\mathbf{X}_k \boldsymbol{\beta}_k^{(r+1)} = \mathbf{H}_k^{(r)} \boldsymbol{\varepsilon}_{ok}^{(r)} \tag{2.25}$$

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

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

and

$$\boldsymbol{\varepsilon}_{0k}^{(r)} = \mathbf{z}_k^{(r)} - \sum_{t=1}^{J_k} \mathbf{Z}_{tk} \boldsymbol{\gamma}_{tk}^{(r+1)} - {\mathbf{W}_{kk}^{(r)}}^{-1} \sum_{s=1, s\neq k}^{p} \mathbf{W}_{ks}^{(r)} \left[\boldsymbol{\eta}_s^{(r+1)} - \boldsymbol{\eta}_s^{(r)} \right]$$

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

A single r^{th} Newton-Raphson step is achieved using a backfitting procedure for each k, cycling through (2.25) and then (2.23) for $j = 1, 2, \ldots, J_k$ and cycling over $k = 1, 2, \ldots, p$ until convergence of the set of updated values $\boldsymbol{\alpha}_k^{(r+1)}$ for $k = 1, 2, \ldots, p$. The updated predictors $\boldsymbol{\eta}_k^{(r+1)}$, 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 $k = 1, 2, \ldots, p$ and $s = 1, 2, \ldots, p$, 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.dist` packages (which is downloaded automatically when (gamlss) is downloaded) 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(x=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, c(0, pNBI(0:39, 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).

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 (2.2), the population probability (density) function $f_Y(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\nu}, \boldsymbol{\tau})$, 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 $f_Y(y|\boldsymbol{\theta})$ 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(`$y \sim 1$`, 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-Exponetial-t͡	`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, $f_Y(y|\boldsymbol{\theta})$, are usually defined on $(-\infty, +\infty)$, $(0, +\infty)$ or $(0, 1)$, but can take other support ranges. Discrete distributions $P(Y = y|\boldsymbol{\theta})$ are usually defined on $y = 0, 1, 2, \ldots, n$, 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.

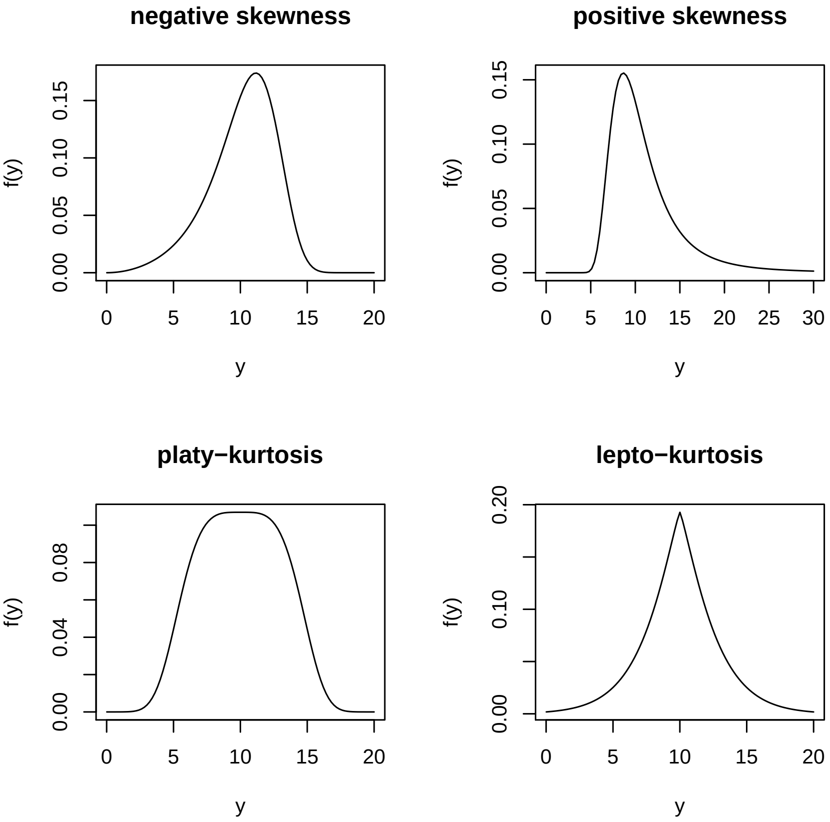

Figure 3.1: Showing different types of continuous distributions

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	`LNO`	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)

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.

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, LOGNO(μ, σ). Let $Y = \exp(Z)$ where $Z \sim$NO(μ, σ), then $Y \sim$LOGNO(μ, σ).

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

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

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

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 SEP1(μ, σ, ν, τ), is defined by assuming Z_1 and Z_2 have power exponential type 2, PE2$\big(0, \tau^{1/\tau}, \tau\big)$, 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 SEP1(μ, σ, ν, τ) distribution for $mu = 0$, $\sigma = 1$, $\nu = 0$ and $\tau = .5, 1, 2, 5, 10, 1000$. Figure 3.3 shows skew SEP1(μ, σ, ν, τ) distributions where $mu = 0$ 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.

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

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

where $k = f_{Y_1}(\mu)/f_{Y_2}(\mu)$ and $I()$ is an indicator variable taking value 1 of 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:

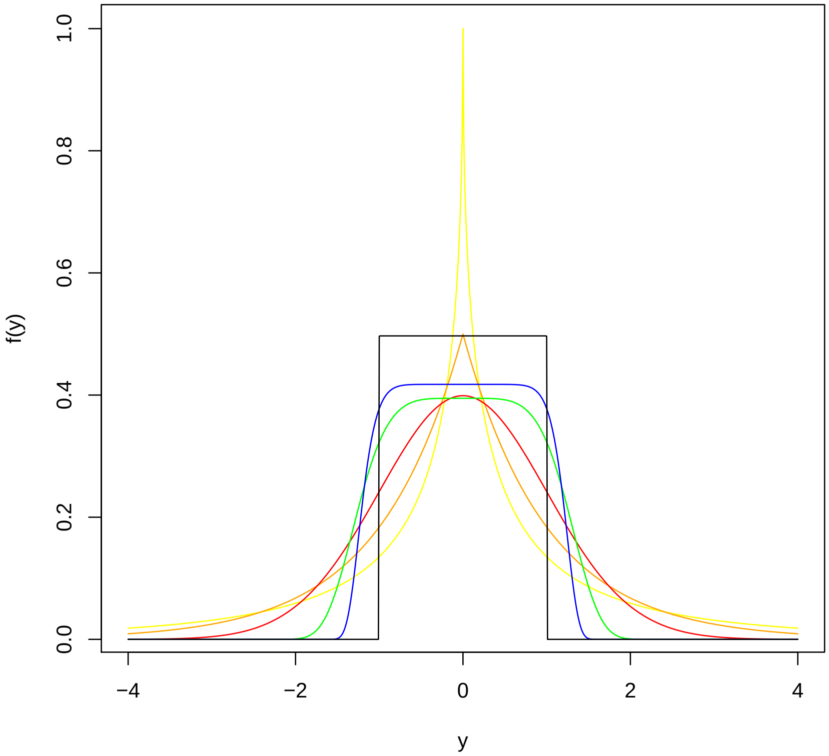

Figure 3.2: Showing different types of continuous distributions

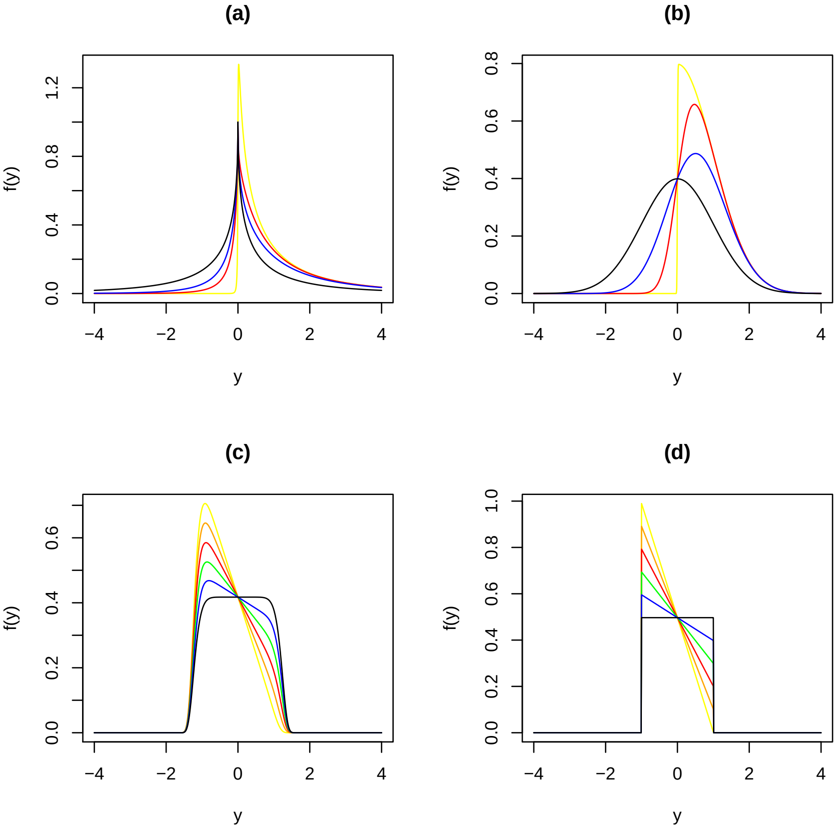

Figure 3.3: Showing different types of continuous distributions

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 $\mathbf{y} = (y_1, y_2, \ldots, y_n)$,

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

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 palace 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:

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

where $F_Y()$ 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

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

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

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

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 $\mu \to y$ and $\sigma \to 0$ 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

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

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 it asymptotic variance is $Var(\hat{\boldsymbol{\theta}}) = I_E(\hat{\boldsymbol{\theta}})$ where $I_E(\hat{\boldsymbol{\theta}})$ is the (Fisher's) expected information $I_E(\boldsymbol{\theta}) = E[d^2\ell/\boldsymbol{\theta}\boldsymbol{\theta}^T]$, 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 Λ 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 non parametric 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 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.data** 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.

```
 **********   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)
```

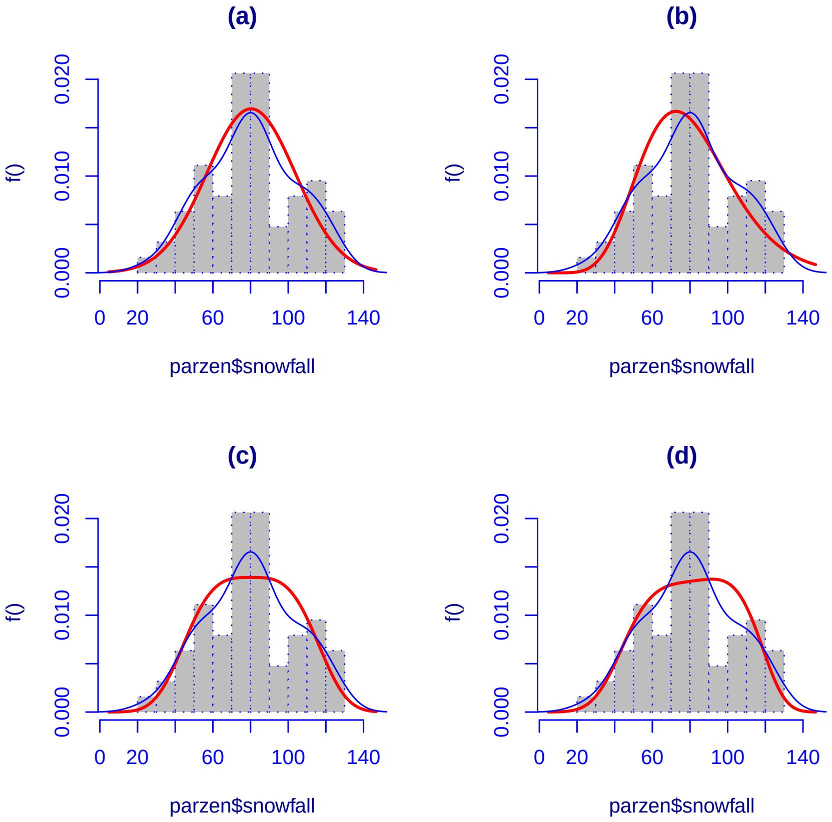

Figure 3.4: Parzen's snowfall data with a kernel density estimate (blue) and fitted (a) normal, (b) gamma, (c) power exponential, and (d) BCPE distributions respectively (red).

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) `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.

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 a explanatory variable exits, 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)

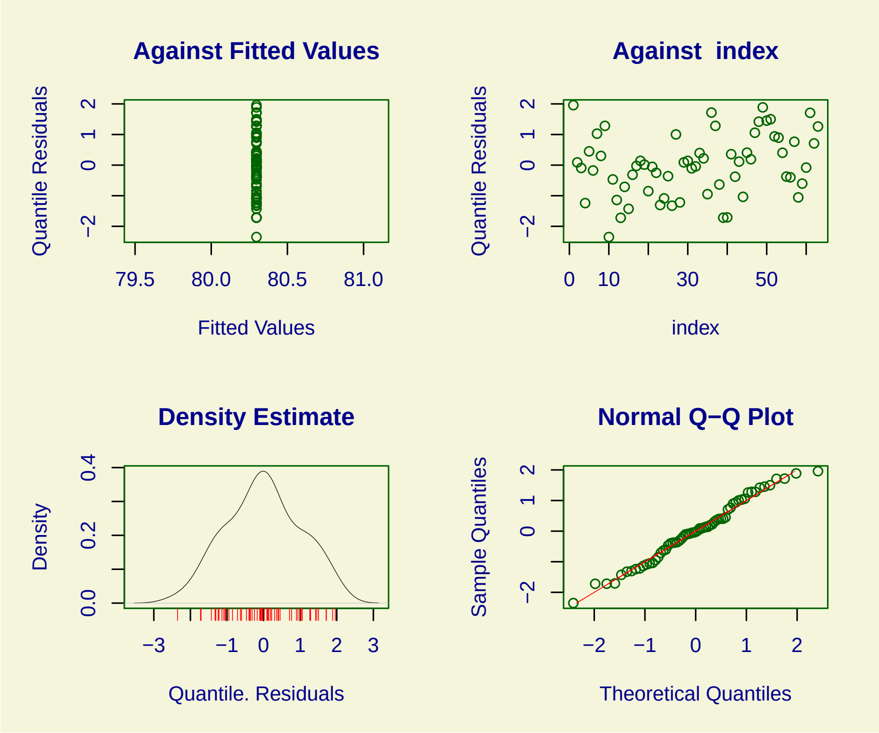

Figure 3.5: The residual plot from the fitted normal model to the snowfall data

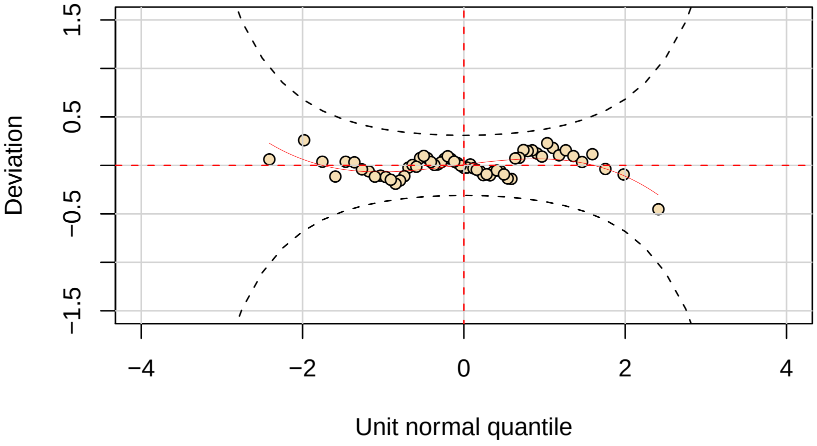

Figure 3.6: The worm plot from the fitted normal model to the snowfall data

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.

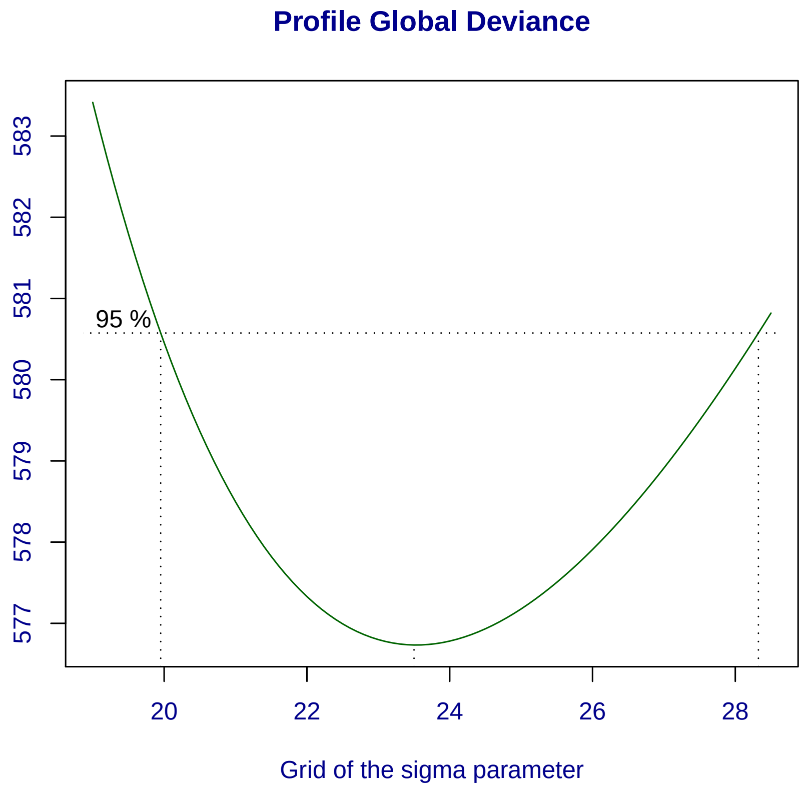

Figure 3.7: The profile deviance for the σ parameter for model modNO

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
```

```
> 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
```

3.6.2 The strengths of glass fibres data

Data summary:

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

sourse: 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:

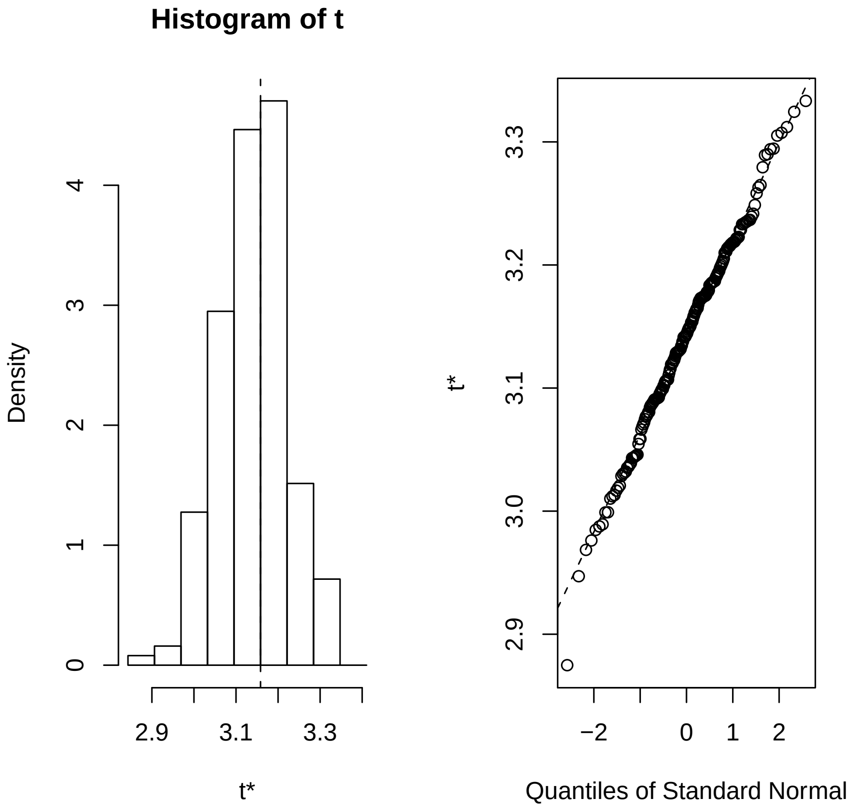

Figure 3.8: The bootstrap for $\log\sigma$ from model modNO

```
> 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 criterio, 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 compate the distribution models the Schwatz Baysian 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
```

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

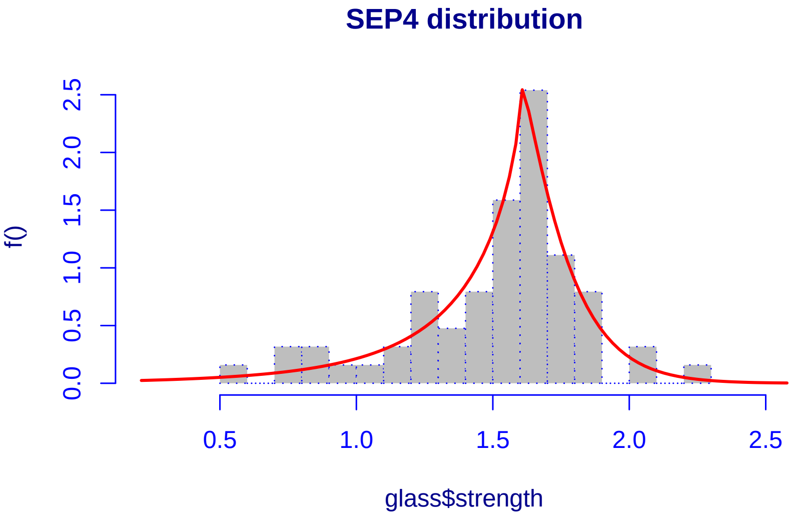

Figure 3.9: The strengths of glass fibres data and the fitted SEP4 distribution model

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)

*******************************************************************
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 he 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.

```
> wp(msep4)
```

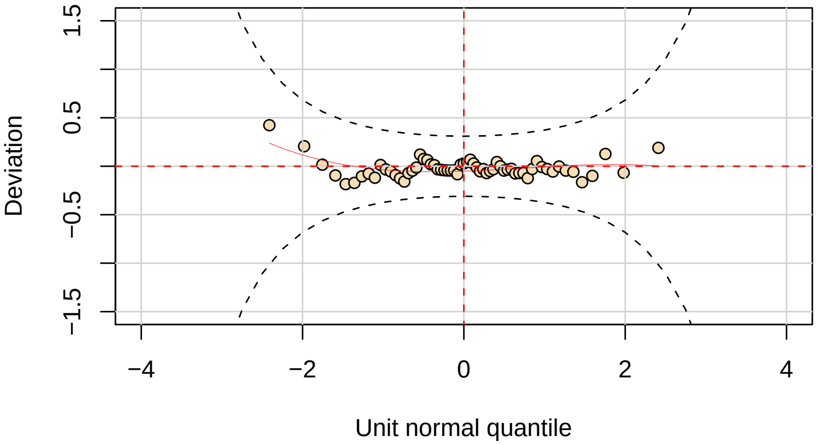

Figure 3.10: A worm plot of the resisuals for the model using the SEP4 distribution fitted to the strengths of glass fibres data.

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

Data summary:

R **data file:** `tensile` in package **gamlss.data** 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 lognormal hypothesis is correct, these data should have been uniformly distributed”. Here we are 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:
 dGAtr pGAtr qGAtr rGAtr GAtr
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)
> mgb1 <- gamlss(str ~ 1, data = tensile, family = GB1, method = mixed(10,
+     100), trace = FALSE)
> mgatr <- gamlss(str ~ 1, data = tensile, family = GAtr, trace = FALSE)
> mlnotr <- gamlss(str ~ 1, data = tensile, family = LOGNOtr, trace = FALSE)
> mtftr <- gamlss(str ~ 1, data = tensile, family = TFtr, trace = FALSE)
> GAIC(mbe, mgb1, mgatr, mlnotr, mtftr)

       df        AIC
mlnotr  2 -3.6714336
mgatr   2 -2.9742870
mbe     2 -2.6101273
mtftr   3 -0.6384319
mgb1    4  0.2258434
```

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, "LOGNOtr" , nbins=10, xlim=c(0,1), main="(a) LOGNOtr")`.

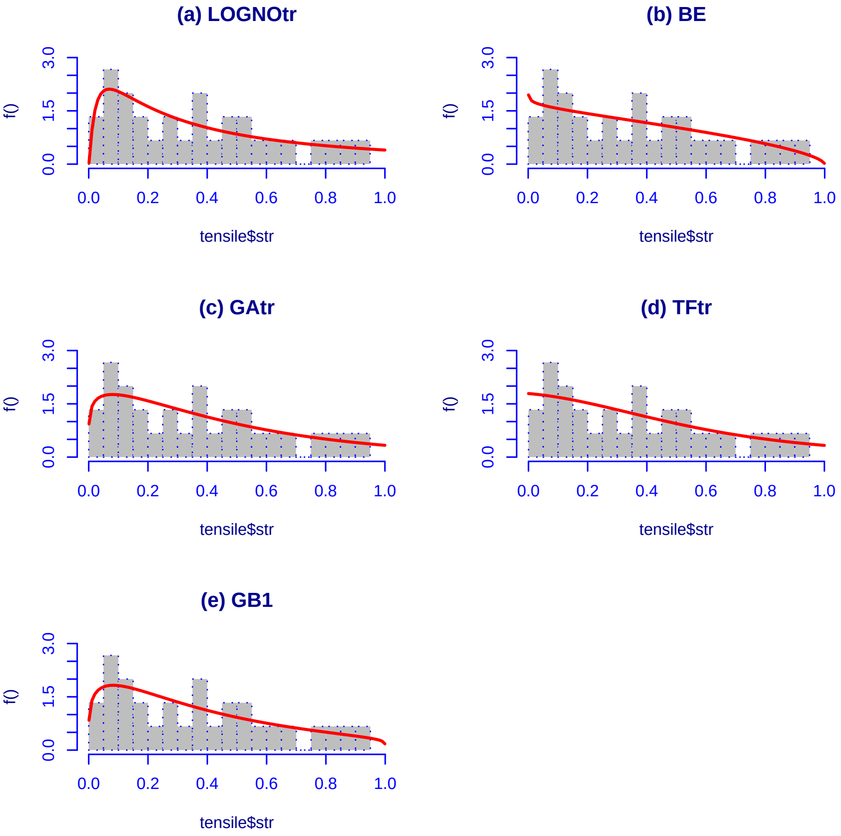

Figure 3.11: Tensile strength data with fitted (a) truncated lognormal , (b) beta, (c) truncated gamma and (d) truncated t (e) generalized beta type 1) distributions.

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
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 BCCG(μ, σ, ν), assumes that Z has a standard normal distribution, NO$(0, 1)$, with mean 0 and standard deviation 1, where

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

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 LNO$\{\mu, \sigma, \nu\}$ where ν is fixed by the user in the GAMLSS software.

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

Table 3.4: Showing distributions generated by univariate transformation

Box-Cox Power Exponential (BCPE)

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

Exponential generalized beta type 2 (EGB2)

The *exponential generalized beta type 2* family for $-\infty < Y < \infty$, denoted by $\mathrm{EGB2}(\mu,\sigma,\nu,\tau)$, 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

$$Z = (\tau/\nu)\exp\left[(Y-\mu)/\sigma\right], \tag{3.9}$$

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 $R = \exp\left[(Y-\mu)/\sigma\right]$ has a beta distribution of the second kind $\mathrm{BE2}(\nu,\tau)$, Johnson *et al.* (1995) p248 and p325 and $B = R/(1+R)$ has an original beta $\mathrm{BEo}(\nu,\tau)$ distribution.

Generalized Beta type 1 (GB1)

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

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

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

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

Note also that $\tau = 1$ in $\text{GB1}(\mu, \sigma, \nu, \tau)$ gives a reparametrization 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 $\text{G3B}(\alpha_1, \alpha_2, \lambda) = \text{GB1}\left(\alpha_1/\left(\alpha_1+\alpha_2\right), \left(\alpha_1+\alpha_2-1\right)^{-1/2}, 1/\lambda, 1\right)$. 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 Lopatatzidis 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 $\mu = \alpha_2\alpha_3^{1/\alpha_1}$, $\sigma = \left(\alpha_1^2\alpha_3\right)^{-1/2}$ 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

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

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

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

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 $v > 0$. 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

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

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

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

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

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 $W \sim \chi^2_\nu \equiv \text{GA}(\nu, [2/\nu]^{1/2})$, 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 $W \sim \chi^2_\tau \equiv \text{GA}(\tau, [2/\tau]^{1/2})$ 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 ST2(μ, σ, ν, τ) 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.

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

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 exGAUS(μ, σ, ν), 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, $NO(\mu, \sigma)$, 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

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

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

Distributions of Y	Distribution of $Y \mid \gamma$	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

Student t family (TF)

The *Student t* family for $-\infty < Y < \infty$, denoted TF(μ, σ, ν), 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).

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

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

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

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

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

Distributions of Y	Distribution of Z_1	Distribution of Z_2	$w(z)$
$\text{SN1}(\mu,\sigma,\nu)$	$\text{NO}(0,1)$	$\text{NO}(0,1)$	νz
$\text{SEP1}(\mu,\sigma,\nu,\tau)$	$\text{PE2}\big(0,\tau^{1/\tau},\tau\big)$	$\text{PE2}\big(0,\tau^{1/\tau},\tau\big)$	νz
$\text{SEP2}(\mu,\sigma,\nu,\tau)$	$\text{PE2}\big(0,\tau^{1/\tau},\tau\big)$	$\text{NO}(0,1)$	$\nu\,(2/\tau)^{1/2}\,sign(z)\lvert z\rvert^{\tau/2}$
$\text{ST1}(\mu,\sigma,\nu,\tau)$	$\text{TF}(0,1,\tau)$	$\text{TF}(0,1,\tau)$	νz
$\text{ST2}(\mu,\sigma,\nu,\tau)$	$\text{TF}(0,1,\tau)$	$\text{TF}(0,1,\tau+1)$	$\nu\lambda^{1/2}z$

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

Skew Normal type 1 (SN1)

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

Skew exponential power type 1 (SEP1)

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, $\text{PE2}\big(0,\tau^{1/\tau},\tau\big)$, distributions in (3.18). Azzalini (1986) called this distribution type I. The skew normal type 1, $\text{SN1}(\mu,\sigma,\nu)$, is a special case of $\text{SEP1}(\mu,\sigma,\nu,\tau)$ obtained by setting $\tau = 2$.

Skew t type 1 (ST1)

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

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

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

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

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

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 $\text{SEP2}(\mu,\sigma,\nu,\tau)$, 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 $w(z) = \nu(2/\tau)^{1/2} sign(z)|z|^{\tau/2}$. Azzalini (1986) developed a reparametrization of this distribution given by setting $\nu = sign(\lambda)|\lambda|^{\tau/2}$ and called it type II. The skew normal type 1, $\text{SN1}(\mu,\sigma,\nu)$, distribution is a special case of $\text{SEP2}(\mu,\sigma,\nu,\tau)$ obtained by setting $\tau = 2$.

Skew t type 2 (ST2)

The *skew t type 2* family, denoted by $\text{ST2}(\mu,\sigma,\nu,\tau)$ 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 $\lambda = (\tau+1)\,/\,\big(\tau + z^2\big)$, 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

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

where $I()$ is an indicator variable taking value 1 of 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

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

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

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

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

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

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 SN2(μ, σ, ν), is defined by assuming $Z \sim NO(0,1)$ 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

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

where $z = (y - \mu)/\sigma$. References to this distribution are given in Johnson *at 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 SEP3(μ, σ, ν, τ), 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, SN2(μ, σ, ν), is a special case of SEP3(μ, σ, ν, τ) given by setting $\tau = 2$.

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$\left(\mu, \sigma 2^{1/\tau}/\nu, \tau\right)$	PE2$\left(\mu, \sigma\nu 2^{1/\tau}, \tau\right)$	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

Skew t type 3 (ST3)

A *skew type 3* distribution for $-\infty < Y < \infty$, Fernandez and Steel (1998), denoted by ST3(μ, σ, ν, τ) is defined by assuming $Z \sim$ TF$(0, 1, \tau) \equiv t_\tau$ in (3.24), or equivalently $Y_1 \sim$ TF$(\mu, \sigma/\nu, \tau)$ and $Y_2 \sim$ 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.

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 $Y_1 \sim$ PE2$\,(\mu, \sigma, \nu)$ and $Y_2 \sim$ 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 $Y_1 \sim$ PE2(μ, σ, ν) and $Y_2 \sim$ PE2$(\mu, \sigma/q, \tau)$ in (3.22), where $q = \Gamma\left[1 + (1/\tau)\right]/\Gamma\left[1 + (1/\nu)\right]$. 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 $Y_1 \sim$ TF(μ, σ, ν) and $Y_2 \sim$ TF(μ, σ, τ) 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 $\nu \le |Z| < \tau$, and a Student t density function for $|z| \ge \tau$, given by

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

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:

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

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 $\left(\sqrt{\beta_1}, \beta_2\right)$ 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 $i = 1, 2, \ldots, 2868$.

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

  ```
  tse<-read.table("C:/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. `mNO<-histDist(tse$ret,"NO")`
  - 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 $i = 1, \ldots, n$, 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: $\mathbf{z}_k = \boldsymbol{\eta}_k + \left[\mathbf{W}_{kk}\right]^{-1} \mathbf{u}_k$, 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 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 $\boldsymbol{\beta}$'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 $\boldsymbol{\theta}^\top = (\theta_1, \theta_2, \theta_3, \theta_4) = (\mu, \sigma, \nu, \tau)$. The RS algorithm can be described as follows:

Algorithm RS (simple)

A simple version of the RS algorithm

- **Start**: Initialize fitted values $\boldsymbol{\theta}_k^{(1,1)}$ for $k = 1, 2, 3, 4$ for distributional parameter vectors of length n, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, $\boldsymbol{\nu}$ and $\boldsymbol{\tau}$ respectively. Evaluate the initial linear predictors $\boldsymbol{\eta}_k^{(1,1)} = g_k\left[\boldsymbol{\theta}_k^{(1,1)}\right]$, for $k = 1, 2, 3, 4$.
- **START OUTER CYCLE** $r = 1, 2, \ldots$ **UNTIL CONVERGENCE**.

  FOR $k = 1, 2, 3, 4$

  - **START INNER CYCLE** $i = 1, 2, \ldots$ **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 $\boldsymbol{\beta}_k^{(r,i)}$.
  - **END INNER CYCLE** on convergence of $\boldsymbol{\beta}_k^{(r,.)}$ and set $\boldsymbol{\beta}_k^{(r+1,1)} = \boldsymbol{\beta}_k^{(r,.)}$, $\boldsymbol{\eta}_k^{(r+1,1)} = \boldsymbol{\eta}_k^{(r,.)}$ and $\boldsymbol{\theta}_k^{(r+1,1)} = \boldsymbol{\theta}_k^{(r,.)}$, 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.data** 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 of the explanatory variable `age` and different distributions.

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.

```
> 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 the response and the explanatory variable(s). One could hope that this would possibly correct some or all of the above problems simultaneously. Figure 4.2 shows plots where several transformations for `cd4` and `age` were tried. A few of the plots in figure 4.2 appear to improve the situation but none of the plots satisfy linearity, homogeneity of variance and a normal error distribution.

```
> 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)
> page <- c("age^-0.5", "log(age)", "age^.5", "age")
> pcd4 <- c("cd4^-0.5", "log(cd4+1)", "cd4^.5")
> for (i in 1:3) {
+     yy <- with(CD4, eval(parse(text = pcd4[i])))
+     for (j in 1:4) {
```

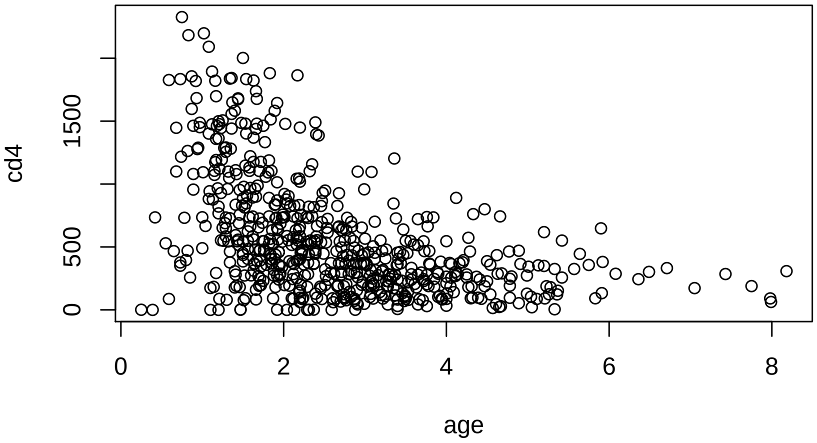

Figure 4.1: The plot of the CD4 data

```
+         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()`):

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

   df      AIC
m7  9 8963.263
```

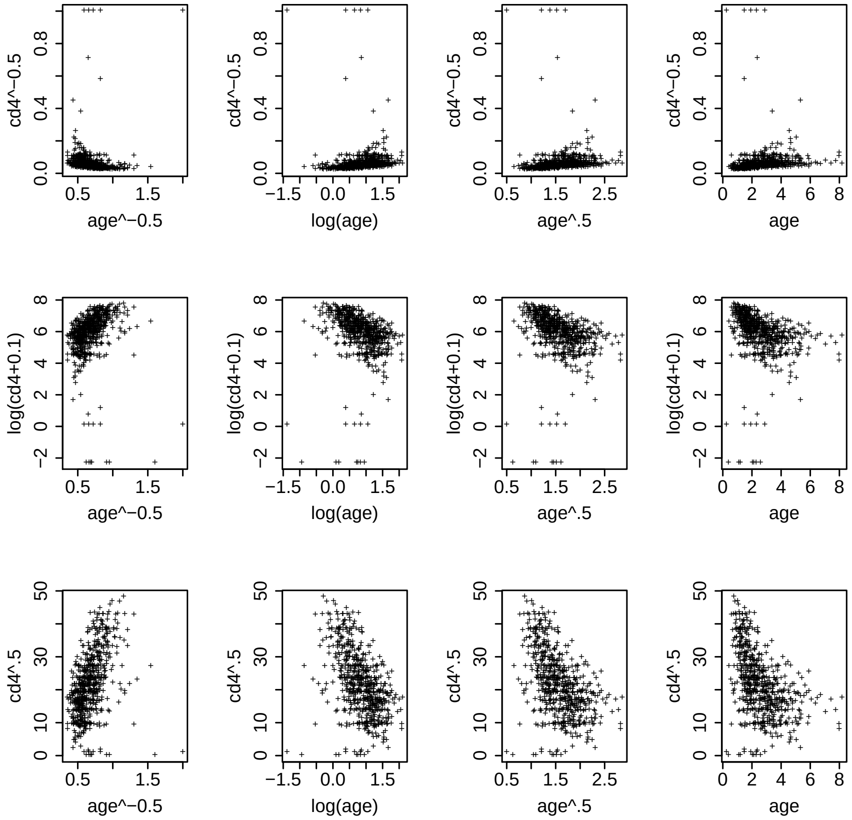

Figure 4.2: The CD4 data with various transformations for CD4 and age

```
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")
```

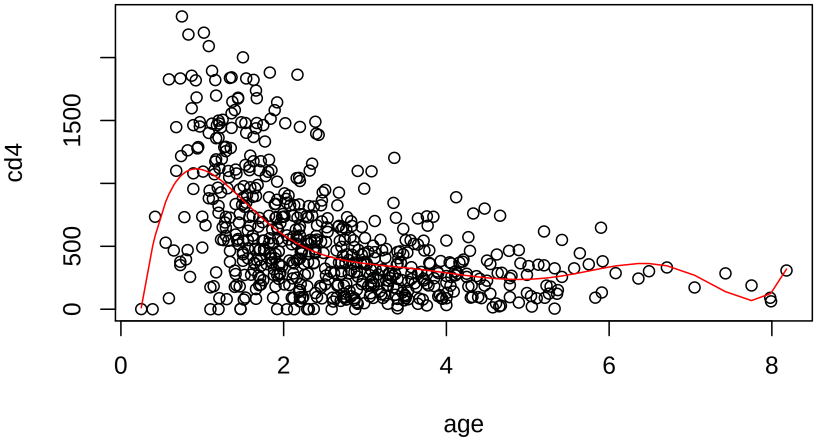

Figure 4.3: The CD4 data and the fitted values using polynomial of degree 7 in age

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 Altmam (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 $age^{-2}\left[\log(age)\right]^2$. Hence the fitted model `m3f` is given by $cd4 \sim NO(\hat{\mu}, \hat{\sigma})$, where $\hat{\mu} = 557.5 - 599.3 + 1116.8\, age^{-2} + 17776.2\, age^{-2}\log(age) + 698.6\, age^{-2}\left[\log(age)^2\right]$ 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
m6b  8 8960.353
m8b 10 8961.073
m5b  7 8964.022
m4b  6 8977.475
m2b  5 8993.351
m3b  5 8993.351
```

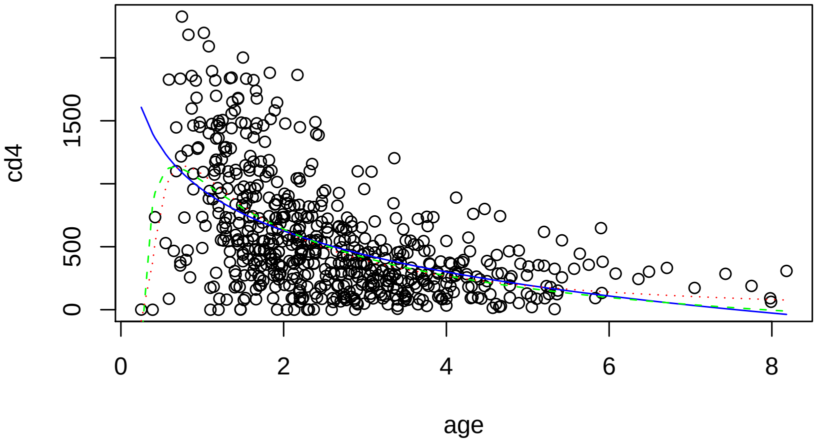

Figure 4.4: The CD4 data and the best fitting fractional polynomials in `age` with one (solid line), two (dashed line) and three (dotted line) terms respectively

```
> 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 SBC uses 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. We first use the function `cs()` of the package **gamlss** introduced in section 2.2.3. The problem with using the smoothing splines function `cs()` is that it does not allow an automatic selection of the smoothing parameter λ. We shall use later the function `pb()` which solves this problem. The smoothing parameter is a function of the (effective) degrees of freedom, so we will use the following procedure using `cs()`: 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:

```
> 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),
```

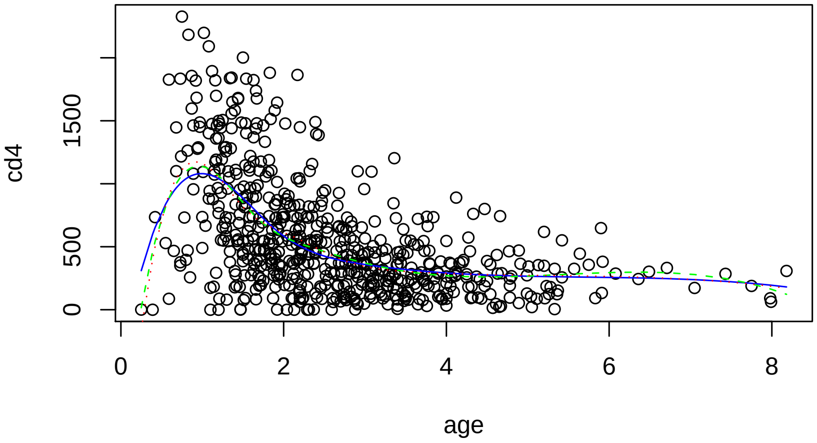

Figure 4.5: The CD4 data and the best fitting piecewise polynomials in age with 5 (___), 6, (− − −) and 7 (. . .), degrees of freedom respectively

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

We now use the function `pb()`. The function `pb()` is a penalized B-spline function as described in section 2.2.3. The `pb()` function automatically selects the smoothing parameter λ and hence the effective degrees of freedom:

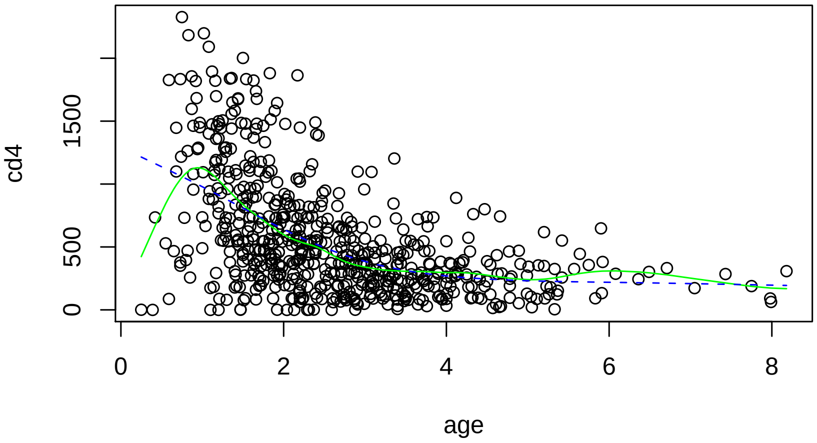

Figure 4.6: The CD4 data and two different cubic spline fits in age, with four (—), and thirteen (− − −) total effective degrees of freedom in the fit

```
> mpb1 <- gamlss(cd4 ~ pb(age), data = CD4, trace = FALSE)
> mpb1$mu.df
```

```
[1] 8.606468
```

The degrees of freedom for μ selected automatically from the function `pb()`, using a local maximum likelihood procedure, are 8.606 (including the 2 for constant and linear term). This is between the values suggested by AIC and SBC above.

We will now use the function `pb()` to find suitable models for both μ and $\log\sigma$ (since the default link function for σ for the normal NO distribution is a log link):

```
> mpb2 <- gamlss(cd4 ~ pb(age), sigma.fo = ~pb(age), data = CD4,
+     gd.tol = 10, trace = FALSE)
> mpb2$mu.df
```

```
[1] 6.200118
```

```
> mpb2$sigma.df
```

```
[1] 3.803651
```

Note that we had to increase the global deviance tolerance level to 10 (the default is 5) in order to allow the model to continue until convergence. The overall degrees of freedom are now 6.2 and 3.8 for μ and σ 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. So by accounting for heterogeneity in the variance we have decreased the complexity of the μ model. Figure 4.7 shows the fitted values for μ for both `mpb1` and `mpb2` models.

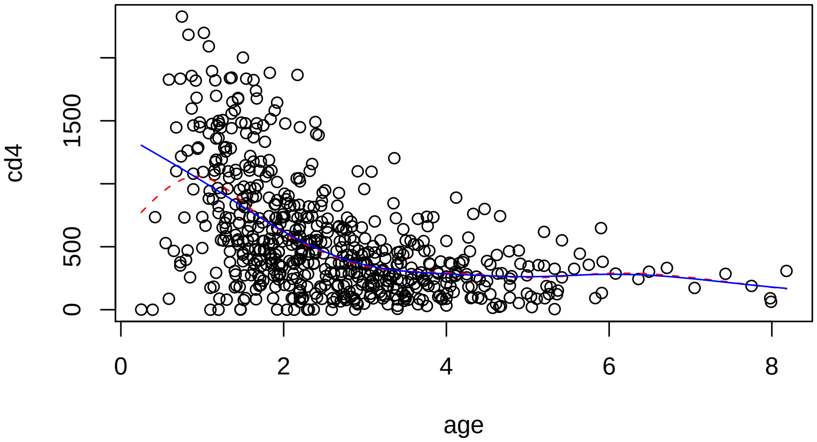

Figure 4.7: The CD4 data and the fitted penelized P-splines in age for model `mpb1` (dashed green line) and `mpb2` (continuous blue line).

Let us consider now what would have happened if we had used the `cs()` function and tried to estimated the degrees of freedom using GAIC.

```
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 total effective degrees of freedom for μ and σ are 5.72 and 3.81 respectively (including constant and linear) very similar to 6.2 and 3.8 achieved using the `pb()` function. Rerunning the code for SBC results in estimated total effective degrees of freedom $(4.55, 2.93)$. 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 that 1. We now fit the AIC chosen model:

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

Figure 4.8 shows the fitted values for the models `m42` and `mpb2` chosen using AIC and `pb()` respectively. The plot is obtained using the command `fitted.plot(m42,mpb2, 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 for any practical purpose identical.

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

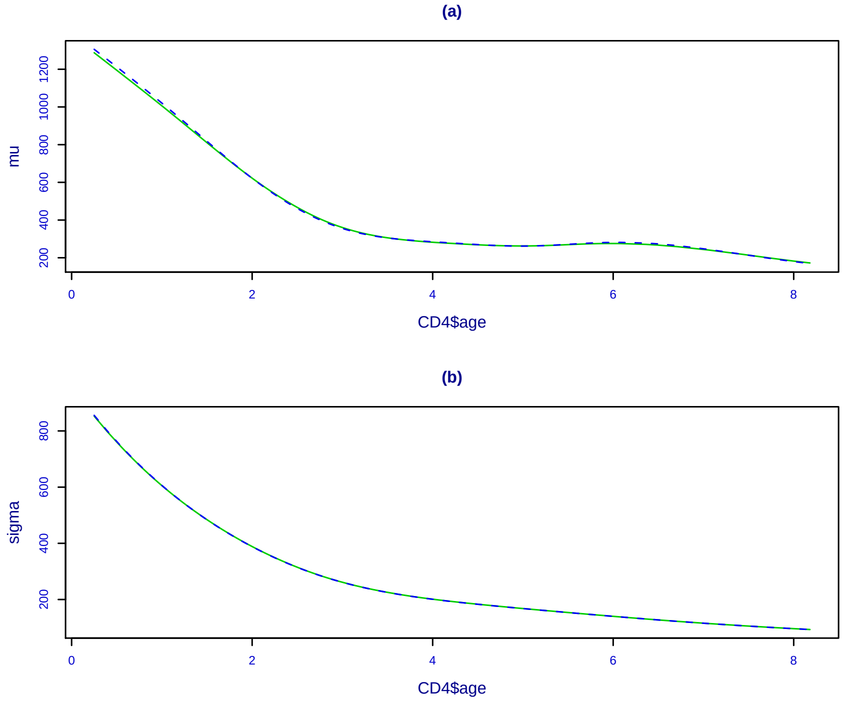

Figure 4.8: A plot of the fitted μ and σ values against `age` for models `m42` (continuous green) and `mpb2` (dashed blue).

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 global 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 total effective degrees of freedom for μ and σ are 6.78 and 3.38 respectively, in this instance not very different from the ones we obtained earlier from using the `pb()` function.

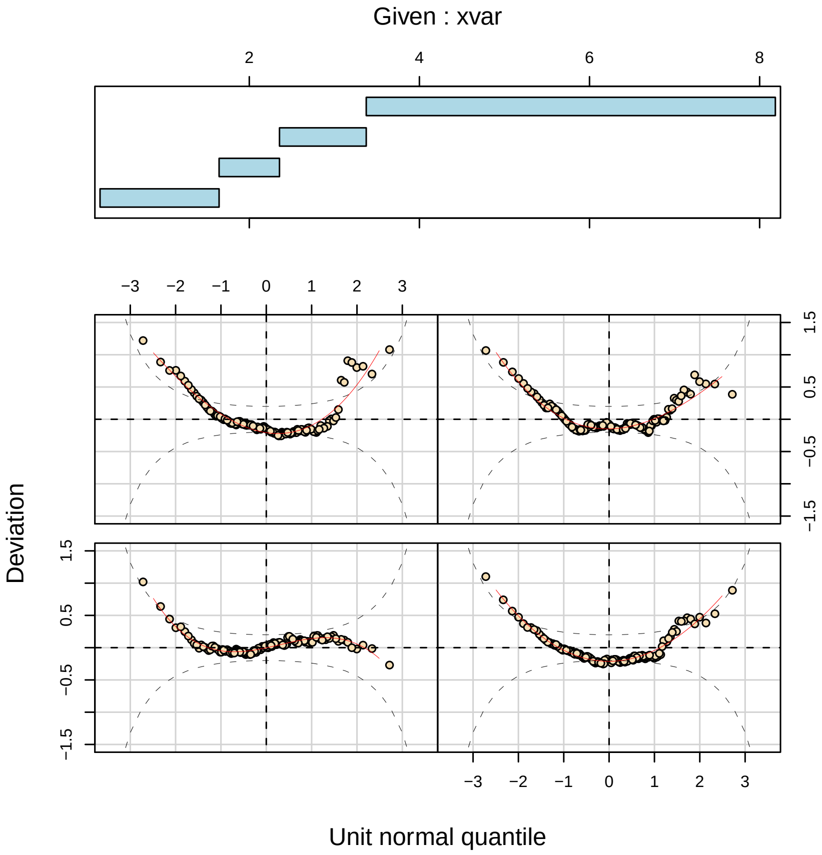

Figure 4.9: A worm plot of the residuals from models `m42`.

Figure 4.9 shows a worm plots from the residuals of model `mpb2`. The worm plot in Figure 4.9 shows four detrended normal Q-Q plots of the (normalized quantile) residuals in four non-overlapping ranges of the explanatory variable `age`. It was produced using the command `wp(mpb2, xvar=CD4$age, ylim.worm=1.5)`. The four ranges are shown above the worm plot. Worm plots were introduced by vanBuuren and Fredriks (2001). The important point here is that quadratic and cubic shapes in a worm plot indicate the presence of skewness and kurtosis respectively in the residuals (within the corresponding range of the explanatory variable, i.e. `CD4$age`). That is, the normal distribution fitted so far to the data is not appropriate. The U shape in the worm plots indicates positive skewness in the residuals (which cannot be modelled by a normal distribution). Also 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. There are a variety of distributions to select from in gamlss including: i) t (TF) ii) power exponential (PE) iii) sinh-arcsinh (SHASH), iv) Jonhson's (JSU) v) five different skew t's (ST1,...,ST5) and vi) four different Skew Power exponential (SEP1,...,SEP4.) distributions. Fitting all these distributions using `pb()` for the predictor model for both μ and σ is a slow process. Table 4.1 presents a AIC table of the different fits. The SEP2 distribution appears to provide the best fit to the data.

distributions	df	AIC
SEP2	8.86	8685.69
SEP1	8.77	8688.08
SEP3	11.24	8693.68
ST2	9.31	8693.91
ST1	9.31	8693.91
ST3	11.33	8694.21
JSU	13.28	8701.05
SHASH	12.75	8705.26
ST5	8.99	8724.25
ST4	12.54	8749.01
TF	11.48	8785.20
PE	11.17	8790.16

Table 4.1: Fitting different distributions to the CD4 data using `pb()` for the predictor models for both μ and σ

4.3 Bibliography

4.4 Exercises for Chapter 4

4.4.1 Exercise 1

Continue the analysis of the abdominal data by fitting different distributions and choosing the 'best' model (as judged by criterion GAIC).

- a) Load the abdominal data and print the variable names:

```
data(abdom)
names(abdom)
```

- b) Fit the normal distribution model using pb() [to fit P-spline smoothers for the predictors for mu and sigma with automatic selection of smoothing parameters]:

```
 mNO<- gamlss(y~pb(x),sigma.fo=~pb(x),data=abdom,family=NO)
```

- c) Try fitting alternative distributions instead of the normal,
  - two parameter distributions: GA, IG, GU, RG, LO
  - three parameter distributions: PE, TF, BCCG
  - four parameter distributions: BCT, BCPE

  MAKE SURE TO USE DIFFERENT MODEL NAMES (instead of mNO)

- d) Compare the fitted models using GAIC with each of the penalties `k=2`, `k=3` and `k=log(length(abdom$y))`, e.g.

```
 GAIC(mNO,mGA,mIG,mGU,mRG,mLO,mPE,mTF,mBCCG,mBCT,mBCPE,mPE2,mTF2,mBCCG2,
      mBCT2,mBCPE2,k=2)
```

- e) For your chosen model look at the total effective degrees of freedom, plot the fitted parameters and plot the data and fitted mu against x:

```
 mLO$mu.df
mLO$sigma.df
fitted.plot(mLO,x=abdom$x)
plot(y~x,data=abdom)
lines(fitted(mLO)~x,data=abdom, col="red")
```

- g) For your chosen model look at the centile curves:

```
centiles(mLO,abdom$x,cent=c(0.4,2,10,25,50,75,90,98,99.6))
```

4.4.2 Exercise 2

For the CD4 count data fit the SEP2 model in the course notes and check it:

- a) Load the CD4 data and print the variable names:

```
data(CD4)
names(CD4)
```

- b) Fit the SEP2 model in the course notes, by first fitting a PE (Power Exponential) model and using its fitted values as starting values for fitting the SEP2 (Skew Exponential Power type 2) model:

```
mPE<-gamlss(cd4~pb(age),sigma.fo=~pb(age),data=CD4,family=PE)
mSEP2<-gamlss(cd4~pb(age),sigma.fo=~pb(age),data=CD4,family=SEP2,
               start.from=mPE,n.cyc=100)
```

- c) For the SEP2 model, print the total effective degrees of freedom and plot the fitted parameters against age.
- d) For the SEP2 model look at the residual and worm plots and Q statistics.
- e) For the SEP2 model look at the centile curves.

4.4.3 Exercise 3

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("C:/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 $0, 1, 2, \ldots, \infty$. 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 difficultly 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 $\Re^+$ then $f_Y(y) = F_W(y+1) - F_W(y)$ is a discrete distribution defined on $y = 0, 1, 2, \ldots, \infty$. Alternatively let $f_Y(0) = F_W(.5)$ and $f_Y(y) = F_W(y + 0.5) - F_W(y - 0.5)$ for $y = 1, 2, \ldots, \infty$. 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 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	$\texttt{PO}(\mu)$	-
Negative binomial type I	$\texttt{NBI}(\mu, \sigma)$	$\texttt{GA}(1, \sigma^{\frac{1}{2}})$
Negative binomial type II	$\texttt{NBII}(\mu, \sigma)$	$\texttt{GA}(1, \sigma^{\frac{1}{2}}/\mu)$
Poisson-inverse Gaussian	$\texttt{PIG}(\mu, \sigma)$	$\texttt{IG}(1, \sigma^{\frac{1}{2}})$
Sichel	$\texttt{SICHEL}(\mu, \sigma, \nu)$	$\texttt{GIG}(1, \sigma^{\frac{1}{2}}, \nu)$
Delaporte	$\texttt{DEL}(\mu, \sigma, \nu)$	$\texttt{SG}(1, \sigma^{\frac{1}{2}}, \nu)$
Zero inflated Poisson	$\texttt{ZIP}(\mu, \sigma)$	$\texttt{BI}(1, 1-\sigma)$
Zero inflated Poisson 2	$\texttt{ZIP2}(\mu, \sigma)$	$(1-\sigma)^{-1}\texttt{BI}(1, 1-\sigma)$
Zero inflated neg. binomial	-	zero inflated gamma
Poisson-Tweedie	-	Tweedie family

Note also that the probability function for the Poisson-Tweedie is given by Hougaarrd *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 that 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 $\Re^+$, 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`(μ)	1	μ	μ
Negative binomial type I	`NBI`(μ, σ)	2	μ	$\mu + \sigma\mu^2$
Negative binomial type II	`NBII`(μ, σ)	2	μ	$\mu + \sigma\mu$
Poisson-inverse Gaussian	`PIG`(μ, σ)	2	μ	$\mu + \sigma\mu^2$
Sichel	`SICHEL`(μ, σ, ν)	3	μ	$\mu + h(\sigma, \nu)\mu^2$
Delaporte	`DEL`(μ, σ, ν)	3	μ	$\mu + \sigma(1-\nu)^2\mu^2$
Zero inflated Poisson	`ZIP`(μ, σ)	2	$(1-\sigma)\mu$	$(1-\sigma)\mu + \sigma(1-\sigma)\mu^2$
Zero inflated Poisson type 2	`ZIP2`(μ, σ)	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 $\sigma \to 0$.

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(μ, σ), 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 $\sigma \to 0$ in the NBI(μ, σ) 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 $\gamma \sim GA(1, (\sigma/\mu)^{\frac{1}{2}})$.

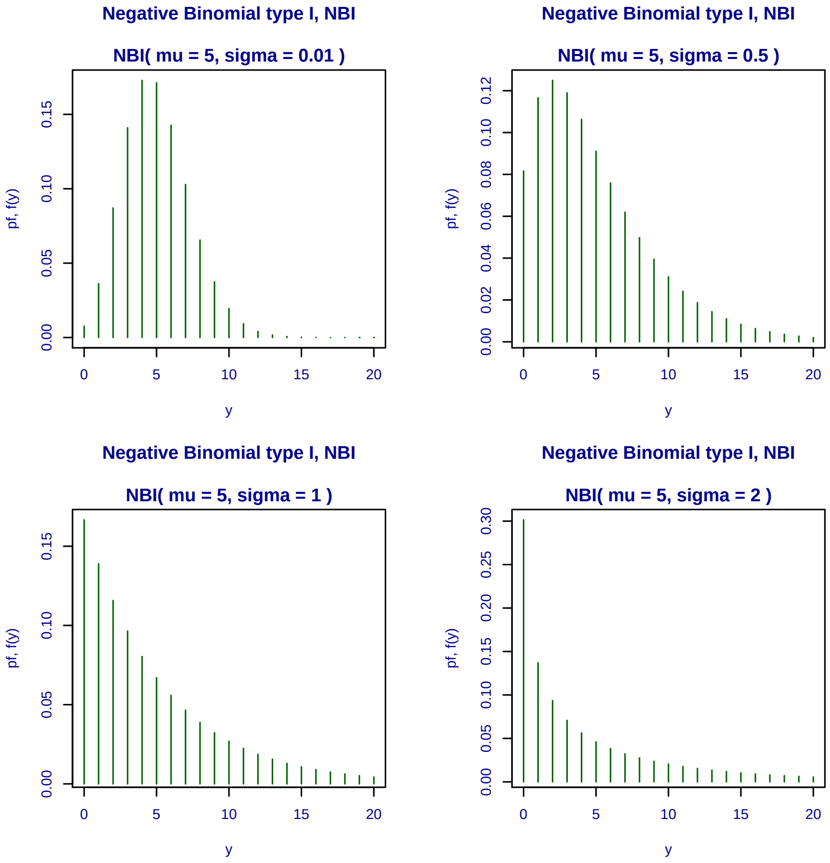

Figure 5.1: NBI distribution for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$

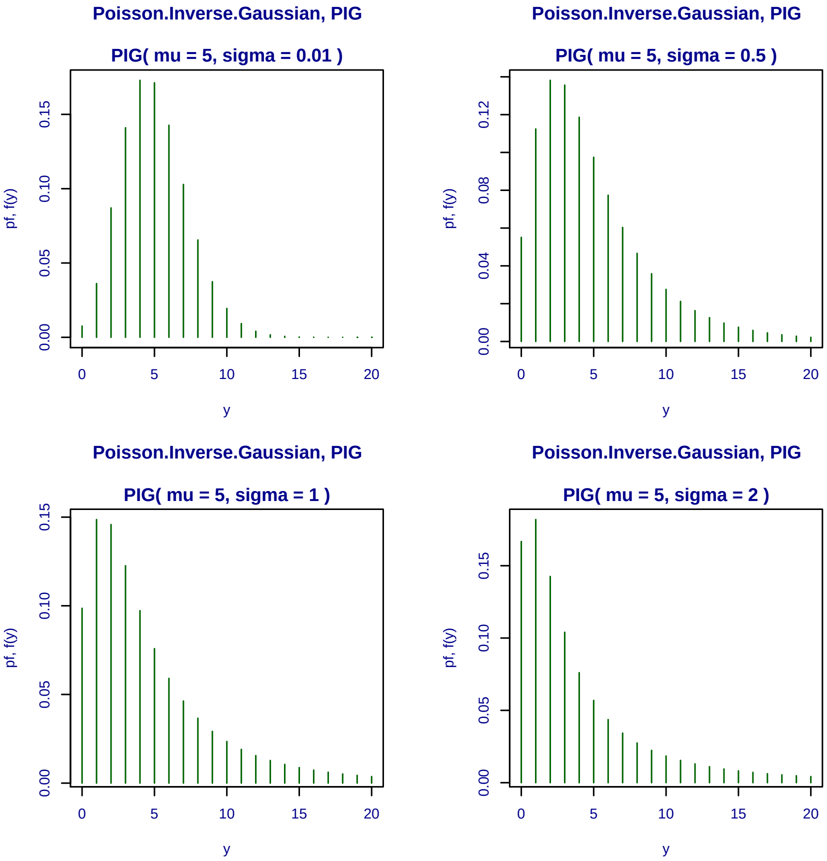

Figure 5.2: PIG distribution for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$

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(μ, σ), 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

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

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

This parameterization of the GIG ensures that $E\left[\gamma\right] = 1$. The mean and variance of Y, are given by $E[Y] = \mu$ and $V(Y) = \mu + \mu^2\left[2\sigma(\nu+1)/c + 1/c^2 - 1\right]$ respectively. For the Sichel distribution $h(\sigma, \nu) = \left[2\sigma(\nu+1)/c + 1/c^2 - 1\right]$ 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

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

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

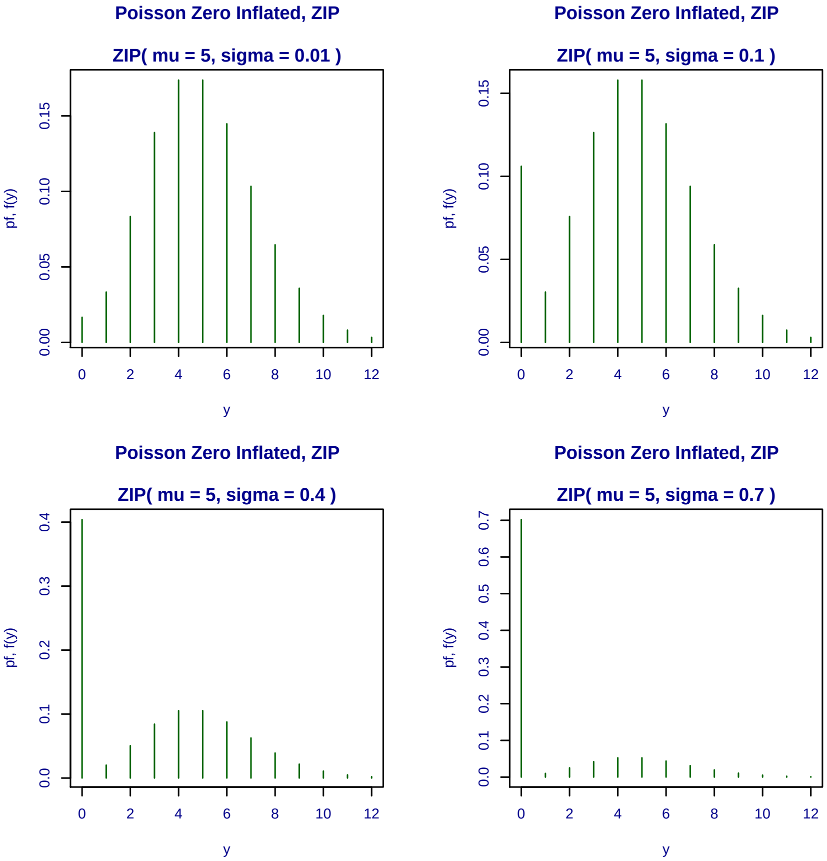

Figure 5.3: ZIP distribution for $\mu = 5$ and $\sigma = (0.01, 0.1, 0.4, 0.7)$

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 $\gamma \sim (1-\sigma)^{-1}\text{BI}(1, 1-\sigma)$. Hence γ has mean 1 and Y has mean μ.

Figure 5.3 plots the zero inflated Poisson distribution, ZIP(μ, σ), for $\mu = 5$ and $\sigma = (0.01, 0.1, .4, 0.7)$. 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="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 Poissson-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 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 $V[\gamma] = \upsilon(\sigma, \nu, \tau)$ 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

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

for $y = 0, 1, 2 \ldots$ 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.

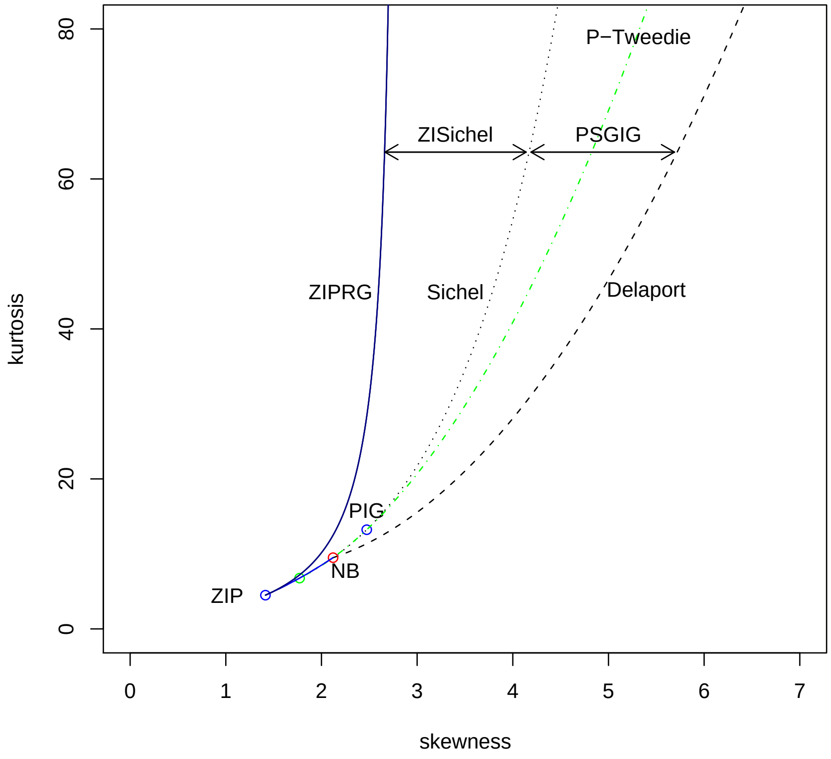

Figure 5.4: Skewness-kurtosis combinations for different distributions for Y (for fixed mean 1 and variance 2 for Y)

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)
```

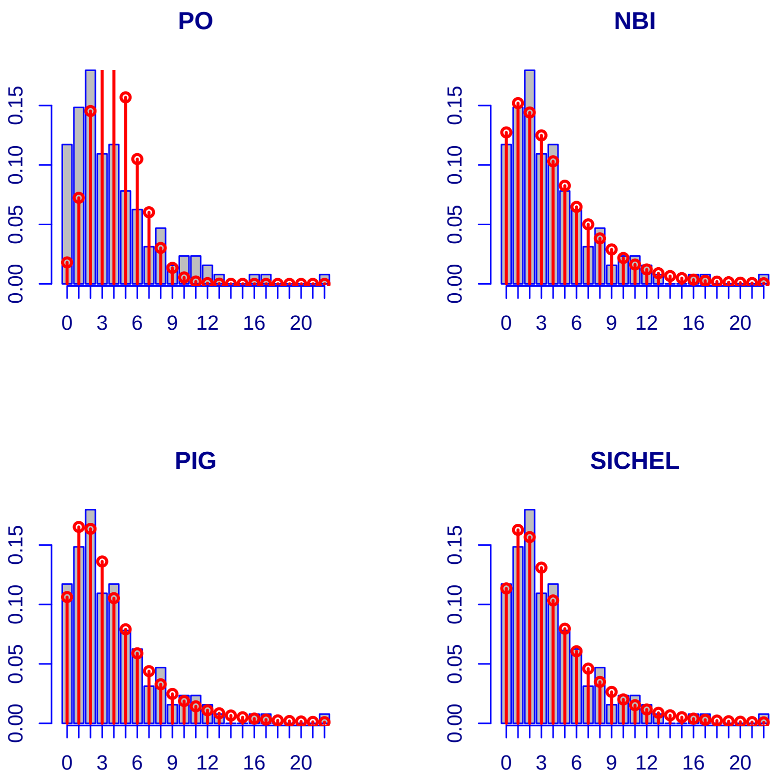

Figure 5.5: The computer failure data fit with (a) Poisson , (b) negative binomial (c) Poisson inverse gaussian and (d) Sichel distributions respectively

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 =head) is given by $Y \sim SICHEL(\hat{\mu}, \hat{\sigma}, \hat{\nu})$ 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)
```

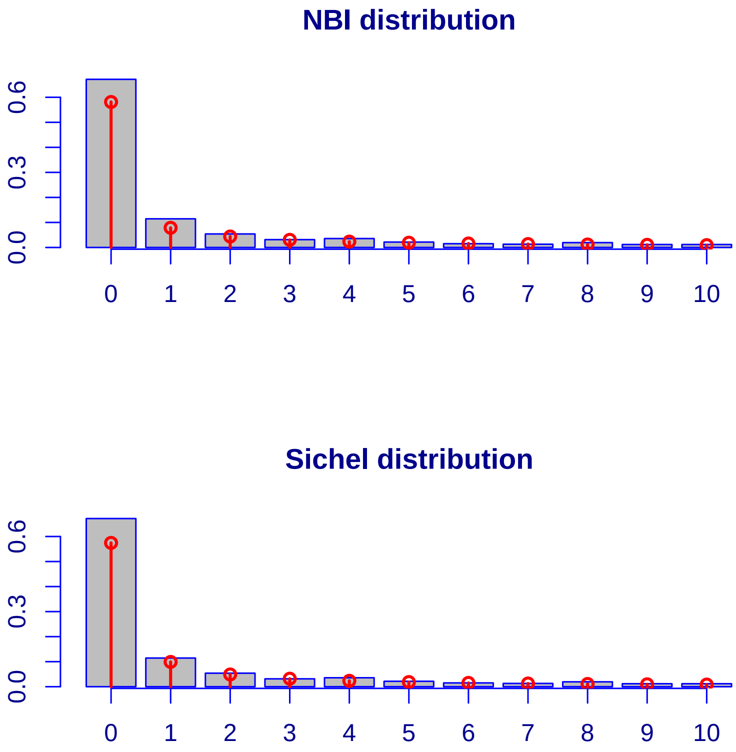

Figure 5.6: The lice data with the fitted (a) negative binomial and (b) Sichel distributions respectively

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 $y = 1, 2, 3, \ldots$, 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 `wold` 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.truc()` 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 that 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:
 dPOtr pPOtr qPOtr rPOtr POtr
The type of truncation is left and the truncation parameter is 0

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

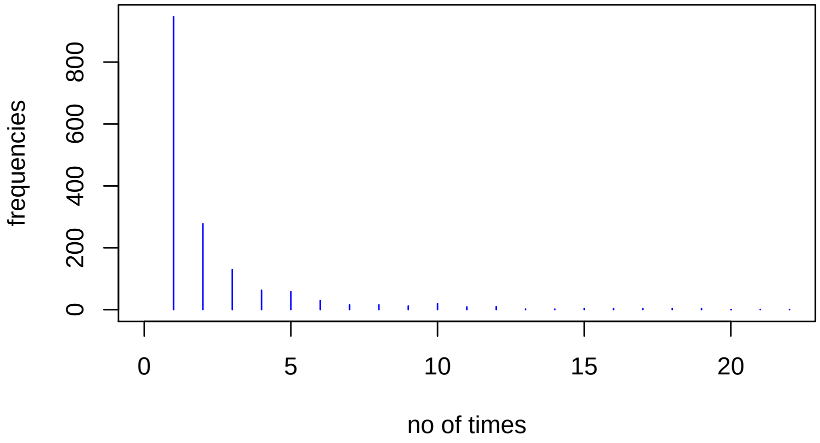

Figure 5.7: The stylometric data: number of time a word appear in a text against the frequencies

```
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 = POtr,
+     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 negaitive 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 distrinution II in part (b) and the Delaporte distribution in (c). The (truncated) Sichel in panel (d) is a superior fit accoriding 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$word, family = POtr, freq = stylo$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:

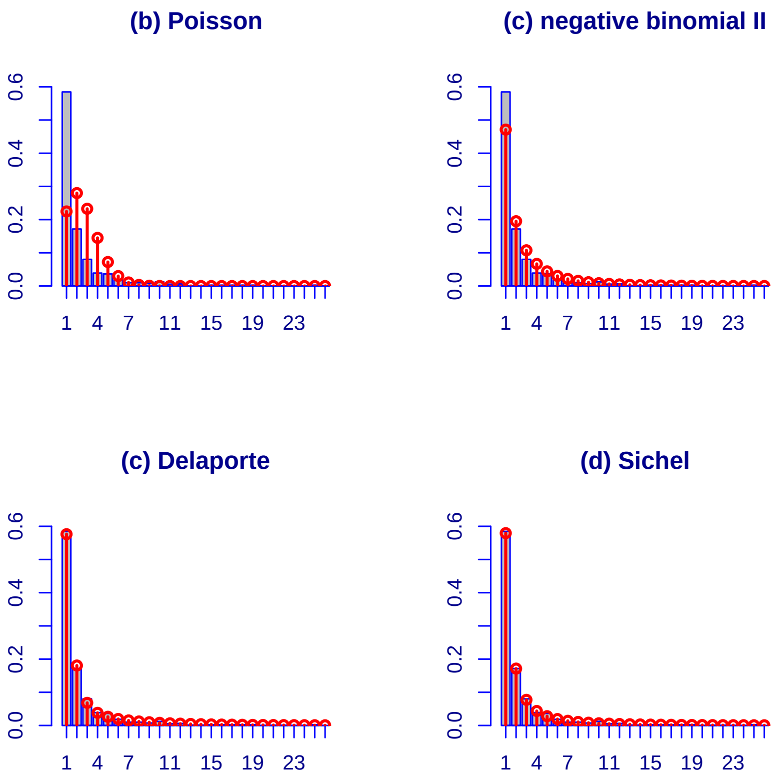

Figure 5.8: The stylometric data fits with (a) Poisson (b) negative binomial type II (c) Delaporte and (d) Sichel distributions respectively

The number of different fish species (`fish`) was recorded for 70 lakes of the world together with explanatory variable `x=log(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)
```

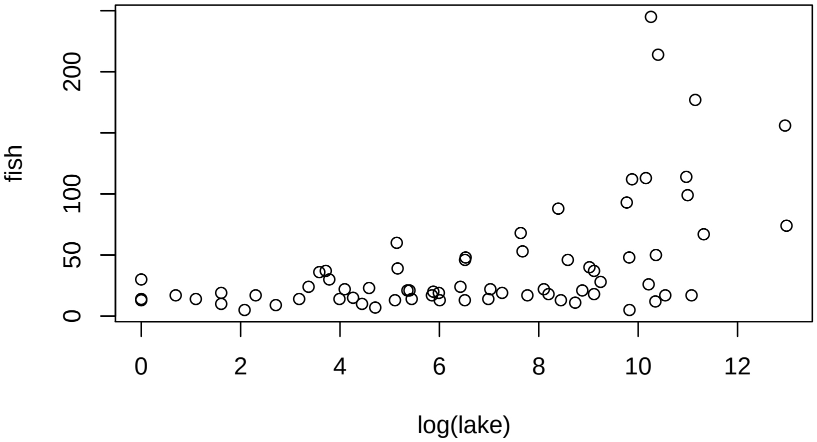

Figure 5.9: The fish species data

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(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 `x` represents a set explanatory variables.

- How does the mean of the response variable depend on `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 `x`?
- Do the scale and shape parameters of the response variable distribution depend on `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(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(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.ql <- list()
> for (i in 1:5) {
+     m.ql[[fam[i]]] <- GAIC(gamlss(fish ~ poly(x, 2), data = species,
+         sigma.fo = ~x, family = fam[i], n.cyc = 60, trace = FALSE))
+ }
> unlist(m.ql)
```

```
     NBI     NBII      PIG      DEL   SICHEL
614.9565 615.1250 612.3684 614.6059 613.7347
```

Modelling log(σ) 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(σ) 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(μ) model rather that 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,

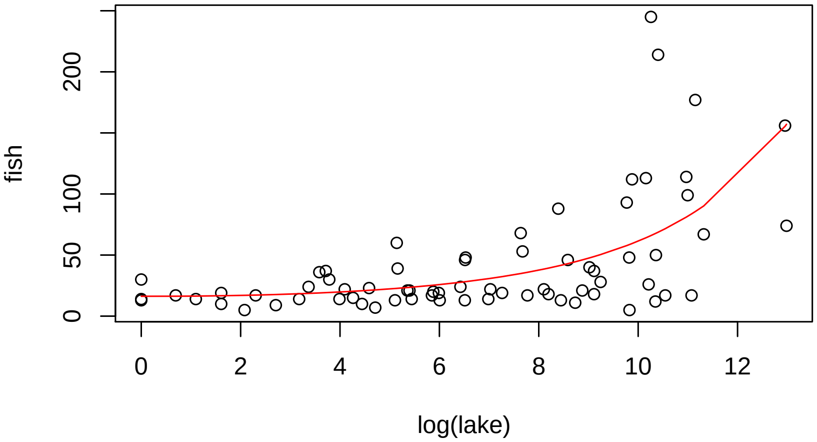

Figure 5.10: Fitted μ (the mean number of fish species) against log lake area

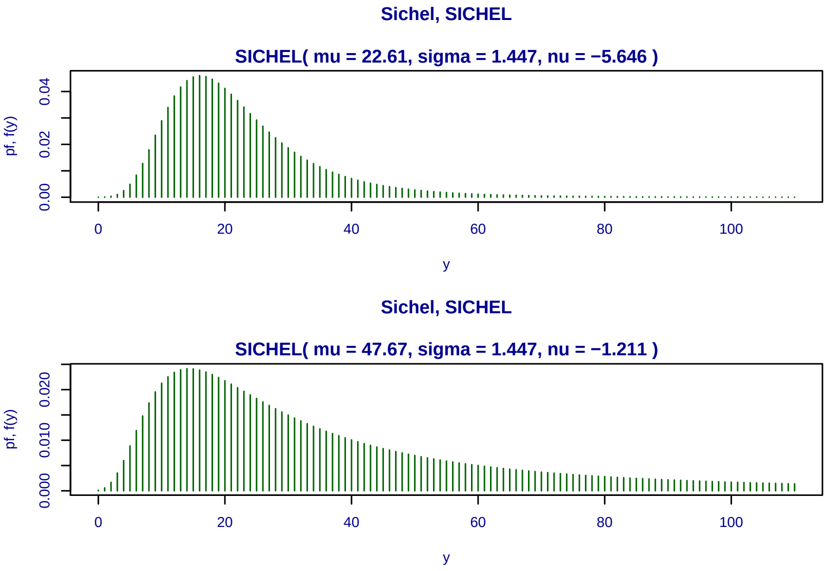

Figure 5.11: Fitted Sichel distributions for observations (a) 40 and (b) 67

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 models was 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 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 Baysien criterion (SBC) the best model is model 17, the discetized 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 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 $Y|\gamma \sim$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

$$K_Y(t) = K_\gamma\left[\mu\left(e^t - 1\right)\right]$$

and hence, assuming that γ has mean 1, the cumulants of Y and γ are related by $E\left(Y\right) = \mu$, $V\left(Y\right) = \mu + \mu^2 V(\gamma,)$,

$$\kappa_{3Y} = \mu + 3\mu^2 V(\gamma) + \mu^3 \kappa_{3\gamma},$$

$$\kappa_{4Y} = \mu + 7\mu^2 V\left(\gamma\right) + 6\mu^3 \kappa_{3\gamma} + \mu^4 \kappa_{4\gamma},$$

where κ_{3Y} and κ_{4Y} are the third and fourth cumulants of Y

The skewness and kurtosis of Y are $\sqrt{\beta_1} = \kappa_{3Y} / \left[V\left(Y\right)\right]^{1.5}$ and $\beta_2 = 3 + \left\{\kappa_{4Y} / \left[V(Y)\right]^2\right\}$ 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 $E(Y) = \mu$, $V(Y) = \mu + \sigma\mu^2$, $\sqrt{\beta_1} = \left(1 + 2\mu\sigma\right) / \left[\mu(1 + \mu\sigma)\right]^{0.5}$ and
$\beta_2 = 3 + \left(1 + 6\mu\sigma + 6\mu^2\sigma^2\right) / \left[\mu(1 + \mu\sigma)\right]$.

Delaporte distribution

If Y has a Delaporte, $DEL(\mu, \sigma, \nu)$, distribution then $E(Y) = \mu$, $V(Y) = \mu + \sigma\mu^2(1-\nu)^2$, $\sqrt(\beta_1) = \mu\left[1 + 3\mu\sigma(1-\nu)^2 + 2\mu^2\sigma^2(1-\nu)^3\right] / \left[V(Y)\right]^{1.5}$ and

$$\beta_2 = 3 + \left\{\mu\left[1 + 7\mu\sigma(1-\nu)^2 + 12\mu^2\sigma^2(1-\nu)^3 + 6\mu^3\sigma^3(1-\nu)^4\right] / \left[V(Y)\right]^2\right\}.$$

Poison-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 $\gamma \sim$ SGIG$(1, \sigma^{1/2}, \nu, \tau)$, defined by $Z = (\gamma - \tau)/(1-\tau) \sim GIG(1, \sigma^{1/2}, \nu)$ and equation 5.1, are given by $E(\gamma) = 1$,

$$V\left(\gamma\right) = (1-\tau)^2 V(Z) = (1-\tau)^2 g_1,$$

$$\kappa_{3\gamma} = (1-\tau)^3 \kappa_{3Z} = (1-\tau)^3\left[g_2 - 3g_1\right],$$

$$\kappa_{4\gamma} = (1-\tau)^3 \kappa_{4Z} = (1-\tau)^4\left(g_3 - 4g_2 + 6g_1 - 3g_1^2\right),$$

where

$$g_1 = \left[1/c^2 + 2\sigma(\nu+1)/c - 1\right],$$

$$g_2 = 2\sigma(\nu+2)/c^3 + \left[4\sigma^2(\nu+1)(\nu+2) + 1\right]/c^2 - 1,$$

$$g_3 = \left[1 + 4\sigma^2(\nu+2)(\nu+3)\right]/c^4 + \left[8\sigma^3(\nu+1)(\nu+2)(\nu+3) + 4\sigma(\nu+2)\right]/c^3 - 1,$$

obtained from $Z = (\gamma - \tau)/(1-\tau) \sim$ GIG$(1, \sigma^{1/2}, \nu)$, where the cumulant generating functions of γ and Z are related by $K_\gamma(t) = \tau t + K_Z\left[(1-\tau)t\right]$.

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

  (a) Fit each of the following distributions for y to the data (using different model names e.g. `mPO` etc. for later comparison): PO(μ), NBI(μ, σ), NBII(μ, σ), PIG(μ, σ), SICHEL(μ, σ, ν), DEL(μ, σ, ν) and ZIP(μ, σ). [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.]

  (b) 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).

  (a) 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)`.

  (b) Fit each of the following distributions for y to the data (using different model names for later comparison): PO (μ), NBI(μ, σ), NBII(μ, σ), PIG(μ, σ), SICHEL(μ, σ, ν), DEL(μ, σ, ν) and ZIP(μ, σ), using factor type for the mean model and a constant scale (and shape).

  (c) Use the AIC command with each of the penalties k = 2, 3 and 4, in order to select a distribution model.

  (d) 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 $y = 0, 1, \ldots, n$, 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 $y = 0, 1, \ldots, n$, where $0 < \mu < 1$, $\sigma > 0$ and n is a known positive integer, has mean $n\mu$ and variance $n\mu(1 - \mu)\left[1 + \sigma(n - 1)/(1 + \sigma)\right]$ 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
```

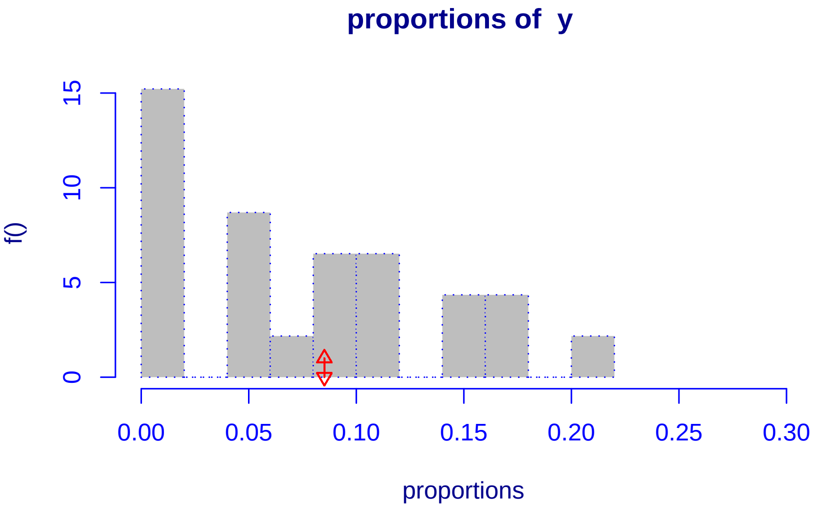

Figure 6.1: The proportion of alveolar-bronchiolar adenomas

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
```

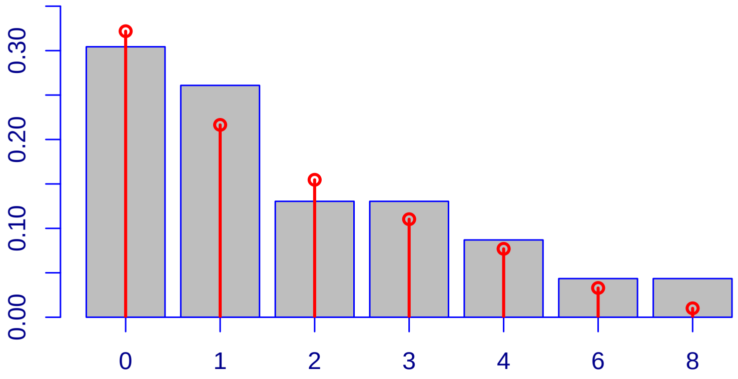

Figure 6.2: The proportion and fitted distribution to the artificial alveolar-bronchiolar adenomas data

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 od 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 if 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 define 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)
```

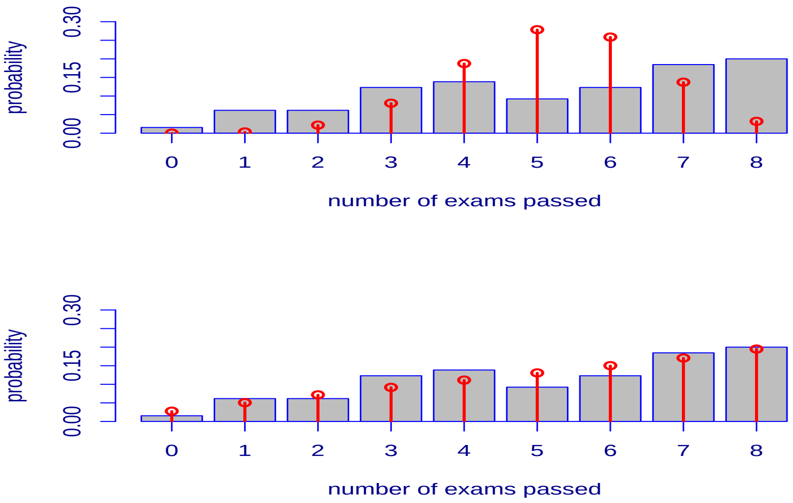

Figure 6.3: The first year student examination results data: Fitted probabilities (a) binomial (b) beta binomial distributions

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

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(μ, σ), with terms `ward`, `year` and `loglos` in the model for `logit`(μ) and term `year` for

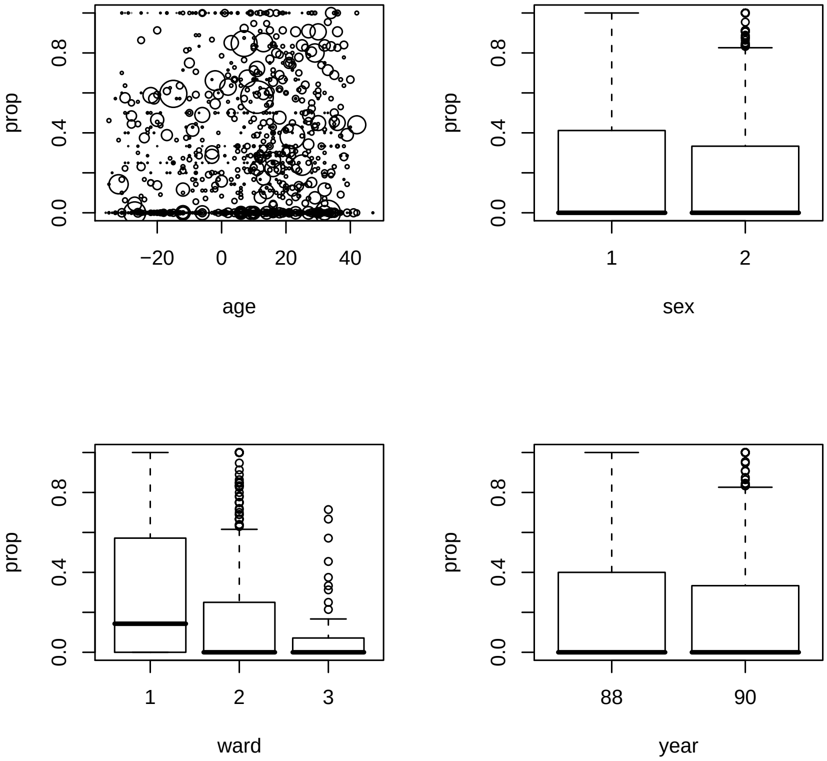

Figure 6.4: The rate of appropriateness against `age`, `sex`, `ward` and `year`

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)

           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 $GD = 4452.36$, $AIC = 4476.36$ aand $SBC = 4539.14$ as shown below:

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]

```
> (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)
```

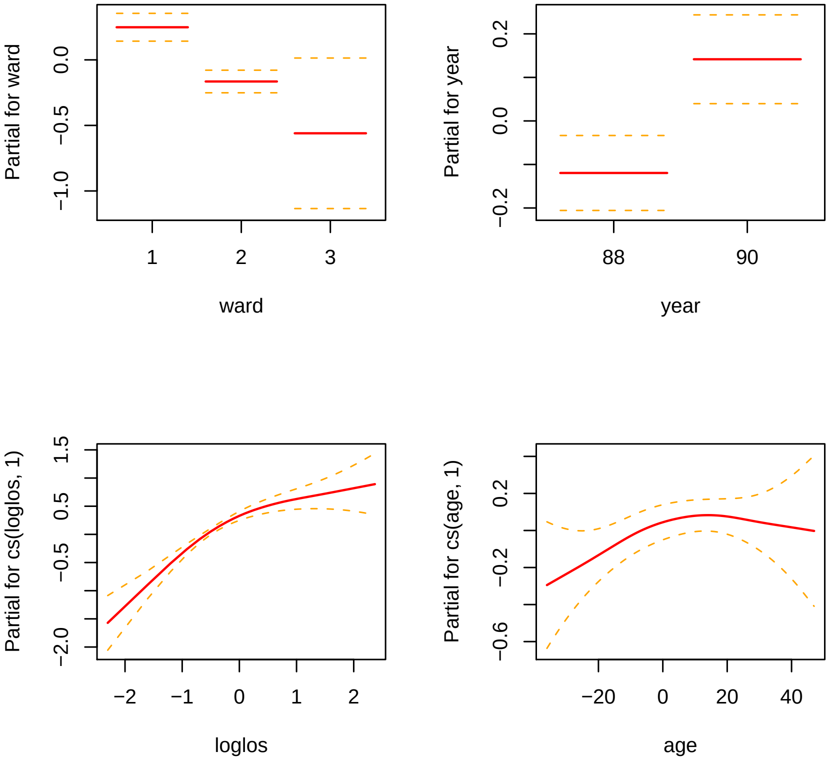

Figure 6.5: The fitted terms for μ in model IV

Figure 6.7 displays six instances of the normalized randomised quantile residuals (see Section 2.2.5) from model IV. The residuals seem to be satisfactory. The figure is generated using the function `rqres.plot()`:

```
> rqres.plot(mIV)
```

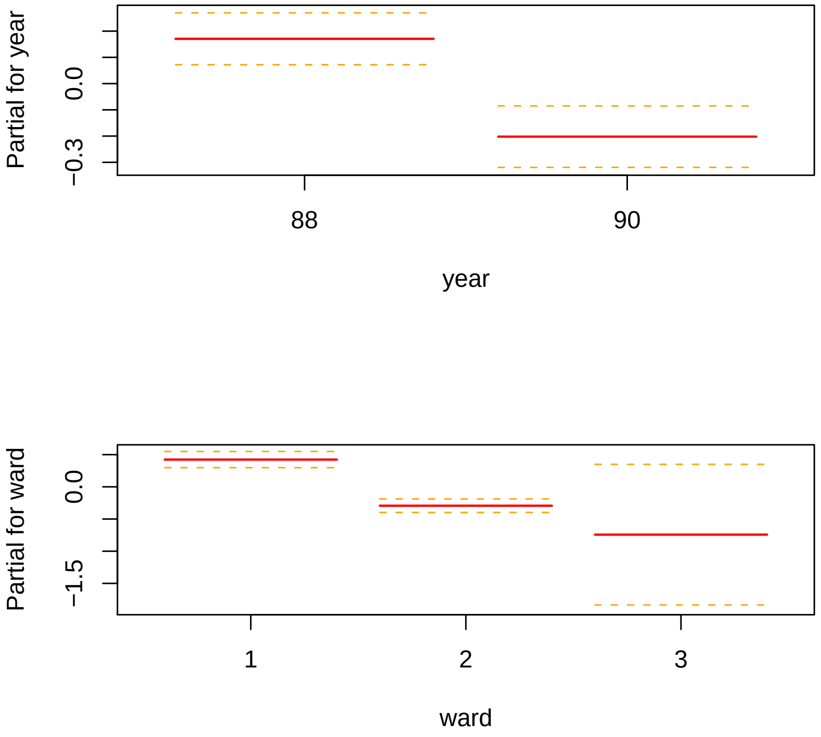

Figure 6.6: The fitted terms for σ in model IV

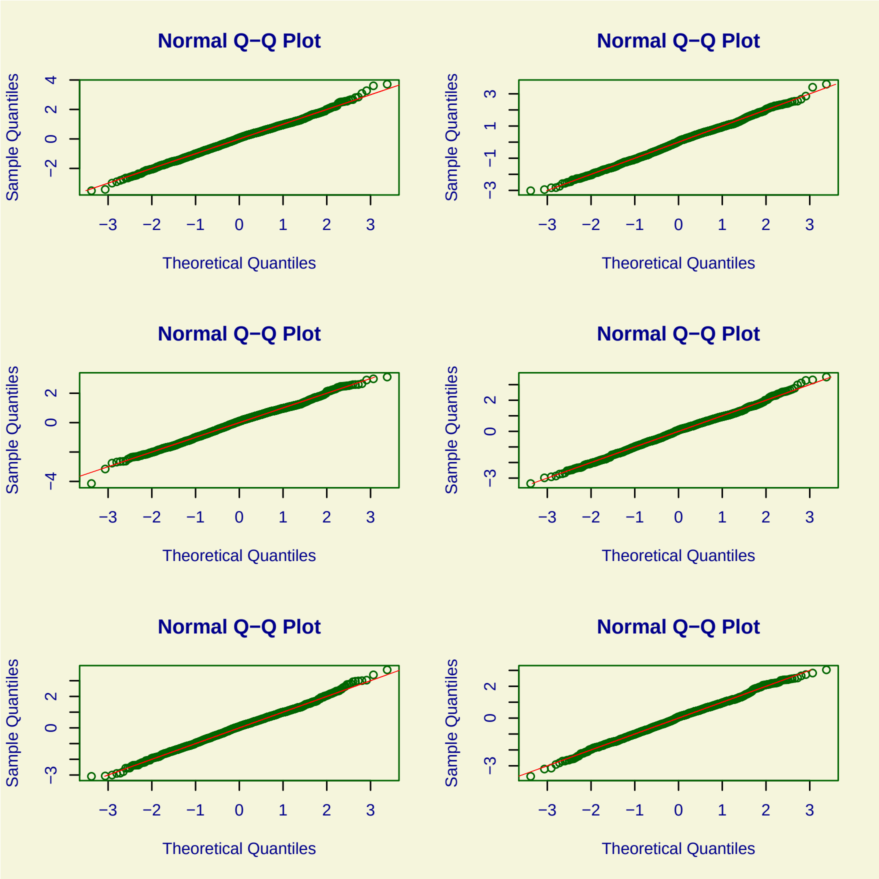

Figure 6.7: Six instances of the normalized randomised quantile residuals for model

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 $k = 1, 2, \ldots, K$, then the (marginal) density of Y is given by

$$f_Y(y) \quad = \quad \sum_{k=1}^{K} \pi_k f_k(y) \tag{7.1}$$

where $0 \leq \pi_k \leq 1$ is the prior (or mixing) probability of component k, for $k = 1, 2, \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$.

More generally the probability (density) function $f_k(y)$ for component k may depend on parameters $\boldsymbol{\theta}_k$ and explanatory variables $\mathbf{x}_k$, i.e. $f_k(y) = f_k(y|\boldsymbol{\theta}_k, \mathbf{x}_k)$.

Similarly $f_Y(y)$ depends on parameters $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\pi})$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ and $\boldsymbol{\pi}^T = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_K)$ and explanatory variables $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K)$, i.e. $f_Y(y) = f_Y(y|\boldsymbol{\psi}, \mathbf{x})$, and

$$f_Y(y|\boldsymbol{\psi}, \mathbf{x}) \quad = \quad \sum_{k=1}^{K} \psi_k f_k(y|\boldsymbol{\theta}_k, \mathbf{x}_k) \tag{7.2}$$

Subsequently we omit the conditioning on $\boldsymbol{\theta}_k$, $\mathbf{x}_k$ and $\boldsymbol{\psi}$ 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 $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k)$ 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 $i = 1, 2, \ldots, n$, from finite mixture model (7.2), the likelihood function is given by

$$L = L(\boldsymbol{\psi}, \mathbf{y}) = \prod_{i=1}^{n} f_{Y_i}(y_i) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k f_k(y_i) \right] \tag{7.3}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, $f_k(y_i) = f_k(y_i|\boldsymbol{\theta}_k, \mathbf{x}_{ki})$, with log likelihood function given by

$$\ell = \ell(\boldsymbol{\psi}, \mathbf{y}) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k f_k(y_i) \right] \tag{7.4}$$

We wish to maximize ℓ with respect to $\boldsymbol{\psi}$, i.e. with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. 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, (Dempser, Laird and Rubin, 1977) to maximize (7.4) with respect to $\boldsymbol{\psi}$, treating all the component indicator variables (i.e $\boldsymbol{\delta}$, defined below) as missing variables.

Let

$$\delta_{ik} = \begin{cases} 1, & \text{if observation } i \text{ comes from component } k \\ 0, & \text{otherwise} \end{cases} \tag{7.5}$$

for $k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, n$. Let $\boldsymbol{\delta}_i^T = (\delta_{i1}, \delta_{i2}, \ldots, \delta_{ik})$ be the indicator vector for observation i. If observation i comes from component k then $\boldsymbol{\delta}_i$ is a vector of zeros, except for the k^{th} value which is $\delta_{ik} = 1$. Let $\boldsymbol{\delta}^T = (\boldsymbol{\delta}_1^T, \boldsymbol{\delta}_2^T, \ldots, \boldsymbol{\delta}_n^T)$ combine all the indicator variable vectors. Then the complete data, i.e. observed $\mathbf{y}$ and unobserved $\boldsymbol{\delta}$, has complete likelihood function given by

$$\begin{aligned} L_c = L_c(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) &= f(\mathbf{y}, \boldsymbol{\delta}) = \prod_{i=1}^{n} f(y_i, \boldsymbol{\delta}_i) \\ &= \prod_{i=1}^{n} f(y_i|\boldsymbol{\delta}_i) f(\boldsymbol{\delta}_i) \\ &= \prod_{i=1}^{n} \left\{ \prod_{k=1}^{K} \left[f_k(y_i)^{\delta_{ik}} \pi_k^{\delta_{ik}} \right] \right\}, \end{aligned} \tag{7.6}$$

since if $\delta_{ik} = 1$ and $\delta_{ik'} = 0$ for $k^{'} \neq k$, then

$$\begin{array}{rcl} f(y_i|\boldsymbol{\delta}_i)f(\boldsymbol{\delta}_i) & = & f_k(y_i)\pi_k, \\ & = & f_k(y_i)^{\delta_{ik}}\pi_k^{\delta_{ik}} \\ & = & \prod_{k=1}^{K} f_k(y_i)^{\delta_{ik}}\pi_k^{\delta_{ik}} \end{array}$$

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

$$\ell_c = \ell_c(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \delta_{ik} \log f_k(y_i) + \sum_{i=1}^{n}\sum_{k=1}^{K} \delta_{ik} \log \pi_k \tag{7.7}$$

If $\boldsymbol{\delta}$ were known then, since $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_K$ 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.

$$\begin{array}{rcl} Q & = & E_{\boldsymbol{\delta}}\left[\ell_c|\mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}\right] \\ & = & \displaystyle\sum_{k=1}^{K}\sum_{i=1}^{n} \hat{w}_{ik}^{(r+1)} \log f_k(y_i) + \sum_{k=1}^{K}\sum_{i=1}^{n} \hat{w}_{ik}^{(r+1)} \log \pi_k \end{array} \tag{7.8}$$

where

$$\begin{array}{rcl} \hat{w}_{ik}^{(r+1)} & = & E\left[\delta_{ik}|\mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}\right] \\ & = & \left[1 * p(\delta_{ik} = 1|\mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)})\right] + \left[0 * p(\delta_{ik} = 1|\mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)})\right] \\ & = & p(\delta_{ik} = 1|\mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}) \\ & = & p(\delta_{ik} = 1|y_i, \hat{\boldsymbol{\psi}}^{(r)}) \\ & = & \dfrac{p(\delta_{ik} = 1, y_i|\hat{\boldsymbol{\psi}}^{(r)})}{f(y_i|\hat{\boldsymbol{\psi}}^{(r)})} \\ & = & \dfrac{f(y_i|\delta_{ik} = 1, \hat{\boldsymbol{\psi}}^{(r)}) \;\; p(\delta_{ik} = 1|\hat{\boldsymbol{\psi}}^{(r)})}{f(y_i|\hat{\boldsymbol{\psi}}^{(r)})} \\ & = & \dfrac{\hat{\pi}_k^{(r)} f_k(y_i|\hat{\boldsymbol{\theta}}_k^{(r)})}{\sum_{k=1}^{K} \hat{\pi}_k^{(r)} f_k(y_i|\hat{\boldsymbol{\theta}}_k^{(r)})} \end{array} \tag{7.9}$$

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 $k = 1, 2, \ldots, K$ 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 $k = 1, 2, \ldots, K$. Assuming, for $k = 1, 2, \ldots, K$, 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 $i = 1, 2, \ldots, n$. Also the parameter $\boldsymbol{\pi}$ only occurs in the second term in (7.8) and so can be estimated by maximimazing 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 $k = 1, 2, \ldots, K$.

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 $k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, n$ to give (7.8).

M-step

(1) for each $k = 1, 2 \ldots, K$, 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 $\mathbf{w}_k^T = (w_{1k}, w_{2k}, \ldots, w_{nk})$,

(2) $\hat{\boldsymbol{\pi}}_k^{(r+1)} = \frac{1}{n} \sum_{i=1}^{n} \hat{w}_{ik}^{(r+1)}$ for $k = 1, 2 \ldots, K$,

(3) $\hat{\boldsymbol{\psi}}^{(r+1)} = \left[\hat{\boldsymbol{\theta}}^{(r+1)}, \hat{\boldsymbol{\pi}}^{(r+1)}\right]$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$.

7.2.3 Modelling the mixing probabilities

Here we extend the finite mixture model by assuming that the mixing probabilities $\boldsymbol{\pi}_k$ for $k = 1, 2, \ldots, K$ for observations $i = 1, 2, \ldots, n$ 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

$$\log\left[\frac{\pi_{ik}}{\pi_{iK}}\right] = \boldsymbol{\alpha}_k^T \mathbf{x}_{0i} \tag{7.10}$$

for $k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, n$. Hence

$$\pi_{ik} = \frac{\exp\left\{\boldsymbol{\alpha}_k^T \mathbf{x}_{0i}\right\}}{\sum_{k=1}^{K} \exp\left\{\boldsymbol{\alpha}_k^T \mathbf{x}_{0i}\right\}} \tag{7.11}$$

for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$ where $\boldsymbol{\alpha}_{\boldsymbol{K}} = \mathbf{0}$. Consequently the complete log likelihood is given by replacing π_k by π_{ik} in equation (7.7) to give

$$\ell_c = \ell_c(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \delta_{ik} \log f_k(y_i) + \sum_{i=1}^{n}\sum_{k=1}^{K} \delta_{ik} \log \pi_{ik} \tag{7.12}$$

This results in replacing π_k with π_{ik} in equations (7.8) and (7.9) of the EM algorithm, i.e.

$$Q \quad = \quad \sum_{k=1}^{K}\sum_{i=1}^{n} \hat{w}_{ik}^{(r+1)} \log f_k(y_i) + \sum_{k=1}^{K}\sum_{i=1}^{n} \hat{w}_{ik}^{(r+1)} \log \pi_{ik} \tag{7.13}$$

where

$$\hat{w}_{ik}^{(r+1)} \quad = \quad \frac{\hat{\pi}_{ik}^{(r)} f_k(y_i|\hat{\boldsymbol{\theta}}_k^{(r)})}{\sum_{k=1}^{K} \hat{\pi}_{ik}^{(r)} f_k(y_i|\hat{\boldsymbol{\theta}}_k^{(r)})} \tag{7.14}$$

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 (7.14), 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 $\hat{\boldsymbol{\theta}}_k^{(r+1)}$, 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 $\hat{\boldsymbol{\alpha}}^{(r+1)}$ 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) $\hat{\boldsymbol{\psi}}^{(r+1)} = \left[(\hat{\boldsymbol{\theta}}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)})\right]$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_K)$.

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 $\hat{\mathbf{w}}_k^T = (\hat{w}_{1k}^T, \hat{w}_{2k}^T, \dots, \hat{\mathbf{w}}_{nk}^T)$ 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.6.5 can be thought of as a finite mixture where the first component is identically zero, i.e. $y = 0$, with probability 1. Hence

$$f_1(y) \quad = \quad \begin{cases} 1, & \text{if y=0} \\ 0, & \text{otherwise.} \end{cases} \tag{7.15}$$

i	k	$\mathbf{y}_e$	$\mathbf{x}_{0e}$	$\mathbf{x}_{1e}$	$\ldots$	$\mathbf{x}_{Ke}$	multinomial response $\mathbf{y}_p$					weights $\hat{\mathbf{w}}^{(r+1)}$
1	1											
2	1	$\mathbf{y}$	$\mathbf{X_0}$	$\mathbf{X}_1$	$\ldots$	$\mathbf{X}_K$	$\mathbf{1}$	$\mathbf{0}$	$\mathbf{0}$	$\ldots$	$\mathbf{0}$	$\hat{\mathbf{w}}_1^{(r+1)}$
⋮	⋮											
n	1											
1	2											
2	2	$\mathbf{y}$	$\mathbf{X_0}$	$\mathbf{X}_1$	$\ldots$	$\mathbf{X}_K$	$\mathbf{0}$	$\mathbf{1}$	$\mathbf{0}$	$\ldots$	$\mathbf{0}$	$\hat{\mathbf{w}}_2^{(r+1)}$
⋮	⋮											
n	2											
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	K											
2	K	$\mathbf{y}$	$\mathbf{X_0}$	$\mathbf{X}_1$	$\ldots$	$\mathbf{X}_K$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\ldots$	$\mathbf{1}$	$\hat{\mathbf{w}}_K^{(r+1)}$
⋮	⋮											
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 setting 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 that 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 2.2.5. The fitted model `m3` appears to be adequate.

```
> plot(m3)
```

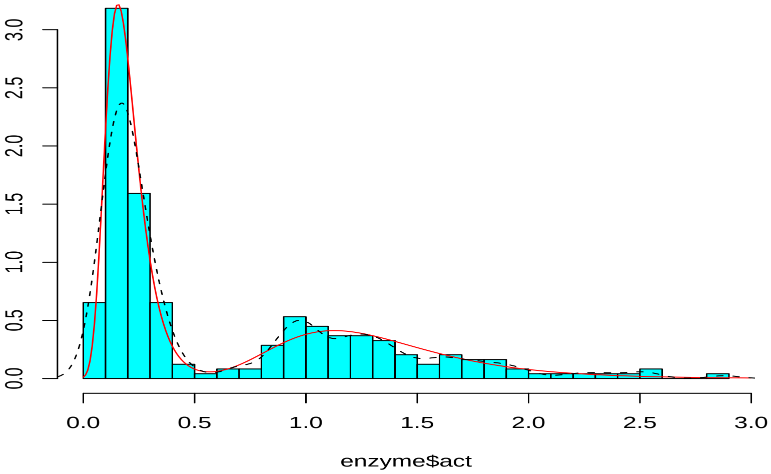

Figure 7.1: A histogram of the enzyme activity data together with a non-parametric density estimator (− − −) and the fitted two component RG model m3 (__).

```
******************************************************************
         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
******************************************************************
```

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.

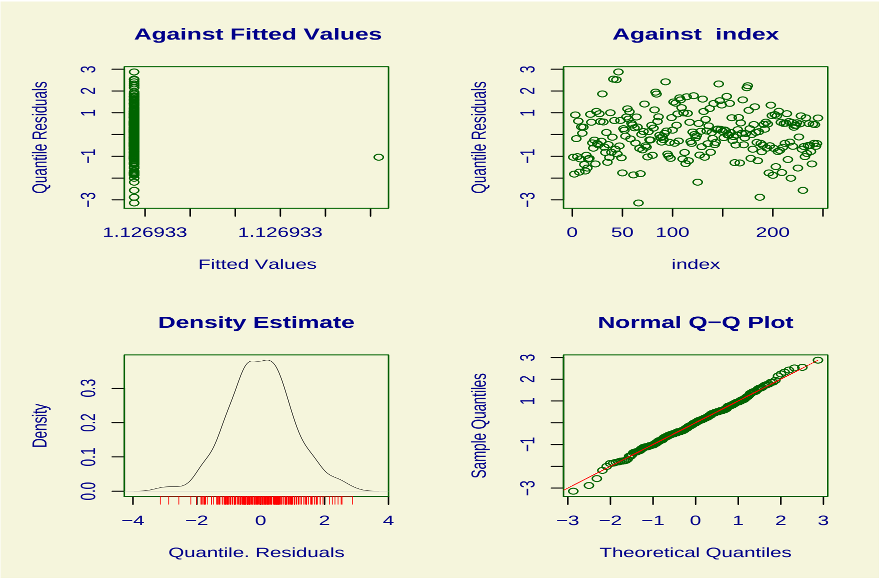

Figure 7.2: The residual plot from the enzyme activity data final two component RG model m3.

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 $f_Y(y) = \hat{\pi}_i f_1(y)+\hat{\pi}_i f_2(y) = 0.67 f_1(y)+0.33 f_2(y)$ 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)
```

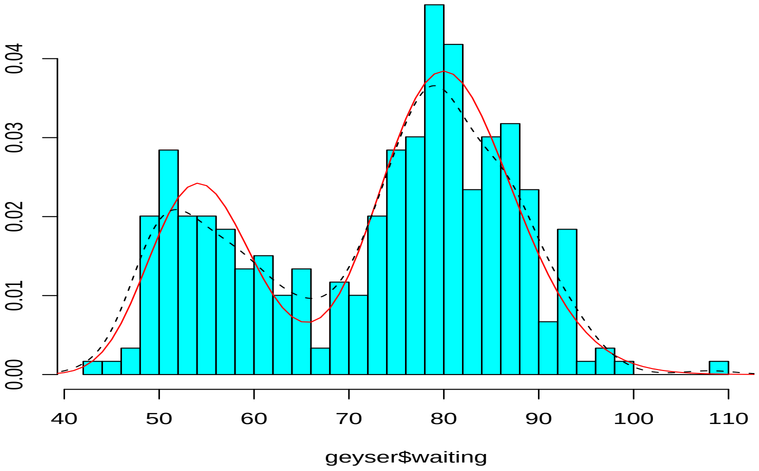

Figure 7.3: A histogram of variable waiting time (to next eruption from the Old Faithful geyser data), together with a non-parametric density estimator (dashed) and the fitted two component IG model (solid).

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
*******************************************************************
```

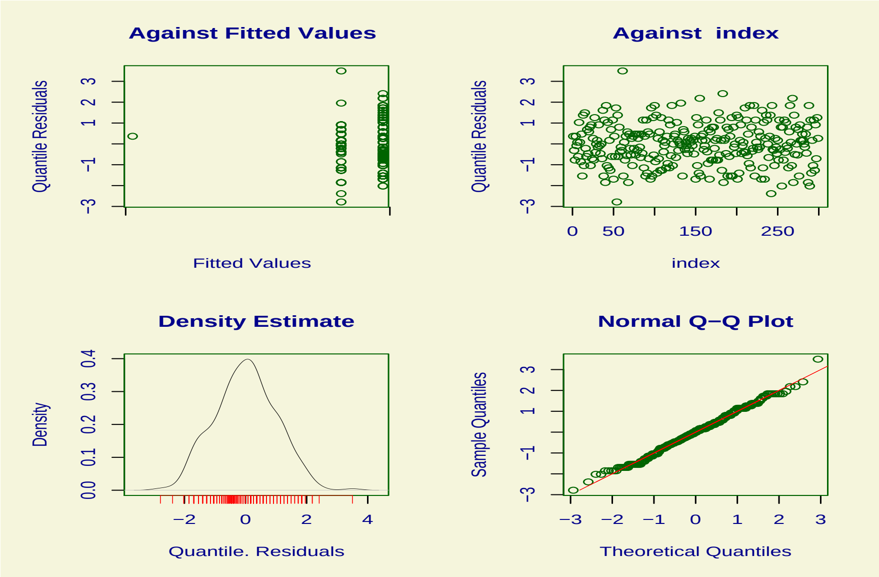

Figure 7.4: The residual plot from the fitted two component IG model for waiting time from the Old Faithful geyser data.

Fitting a finite mixture to a simple regression problem

Data summary: the old faithful geyser

R **data file:** `geyser2` created from `gayser` 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
mNO4  8 1936.679
mNO3  7 1953.317
mIG3  7 1961.234
mIG2  6 1970.647
mNO2  6 1981.932
mIG1  5 2315.304
mNO1  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

$$\log\left[\hat{\pi}_2/(1-\hat{\pi}_2)\right] = \boldsymbol{\eta}_\pi = 10.1892 - 3.1318 * \text{duration}$$

.

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)
```

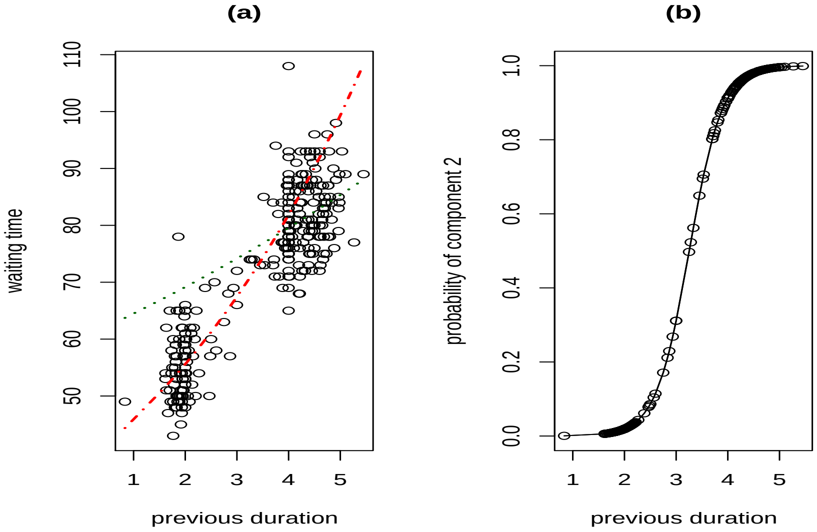

Figure 7.5: (a) A scatter plot of the waiting time against the previous eruption duration from the Old Faithful geyser data together with the fitted values from the two componets, (dotted and dashed for component 1 and 2 respectively) (b) a plot of the probability of belonging to component 2 as a function of duration estimated from model `mIG4` .

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))
> etapi <- 10.19069 - 3.132215 * grid$duration
> etamu1 <- 4.09618 + 0.07007 * grid$duration
> 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)
```

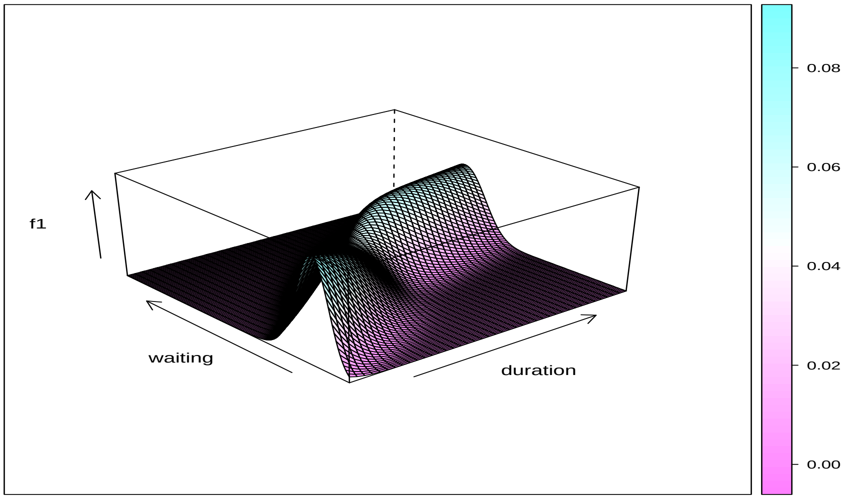

Figure 7.6: Fitted conditional probability density function (`f1`) for waiting time given previous eruption duration for model `mIG4`.

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 that 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)
```

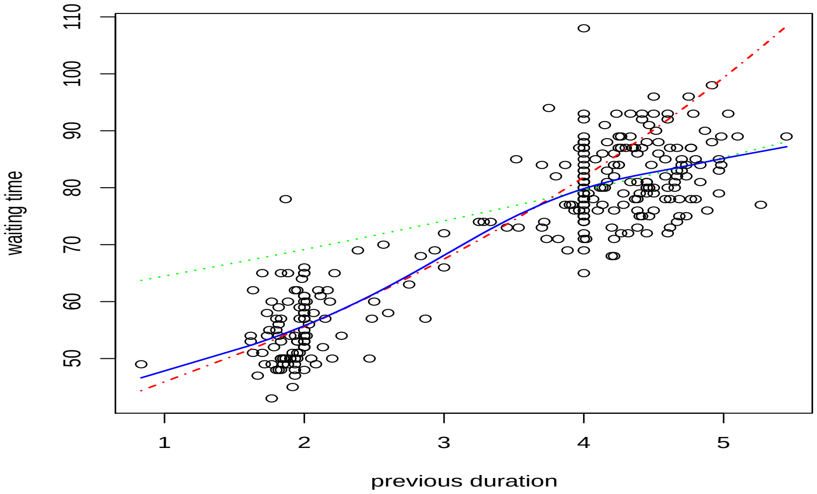

Figure 7.7: Comparison of the fitted values for μ for models mIG4 (dashed and dotted lines) and mIG6 (solid line).

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")$col), main = "(a)"),
+     split = c(1, 1, 2, 1), more = TRUE)
> print(levelplot(f2 ~ duration * waiting, data = grid, colorkey = F,
+     at = seq(0, 0.075, 0.001), xlab = "previous duration", ylab = "waiting time",
+     col.regions = rev(trellis.par.get("regions")$col), main = "(b)"),
+     split = c(2, 1, 2, 1))
```

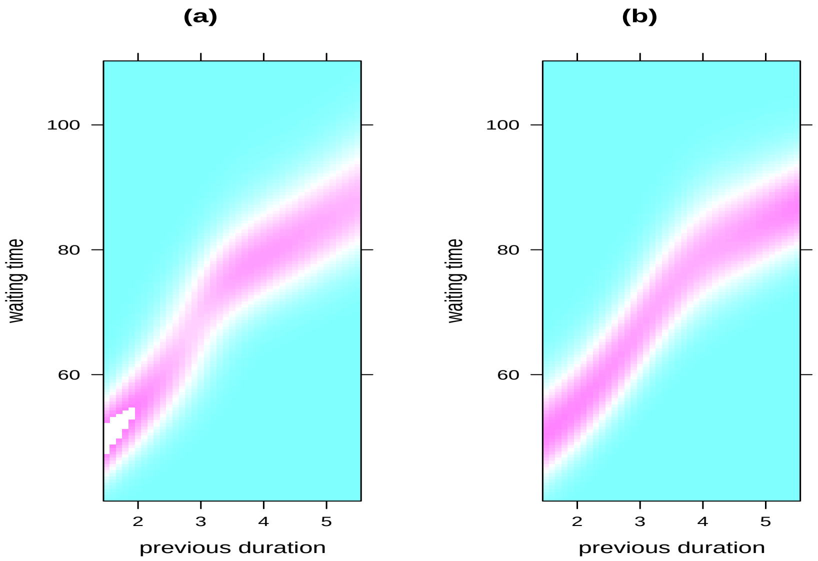

Figure 7.8: Levelplot of the fitted conditional probability density function of the waiting time given the previous eruption time for models (a) mIG4 and model (b) mIG6.

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 $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k)$ 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 $\boldsymbol{\alpha}$ through a multinomial logistic model as in Section 7.2.3. We assume that the K components $f_k(y) = f_k(y|\boldsymbol{\theta}_k, \mathbf{x}_k)$ for $k = 1, 2, \ldots, K$ 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 $\boldsymbol{\theta}_k$ 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 $\log \mu_k = \beta_{ok} + \beta_1 x$. 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_{ok} 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 $\log \sigma_k = \log \sigma = \beta_{2o} + \beta_{21} x$. 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 $\log \mu_k = \beta_{1ok} + \beta_{11k} x$, $\log \sigma_k = \beta_{2ok} + \beta_{21k} x$, $\nu_k = \nu = \beta_{3o}$ and $\log \tau_k = \log \tau = \beta_{4o}$. Here predictor parameters β_{1ok} 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_{3o} for ν and β_{4o} 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 (7.14) 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 $\boldsymbol{\theta}_k$. Obtain $\hat{\boldsymbol{\theta}}^{(r+1)}$ 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 $\hat{\boldsymbol{\alpha}}^{(r+1)}$ by fitting a multinomial logistic model as in Section 7.2.3.

(3) $\hat{\boldsymbol{\psi}}^{(r+1)} = \left[\hat{\boldsymbol{\theta}}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}\right]$

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

i	MASS	$\boldsymbol{y}_e$	$\mathbf{X}_e$	$\hat{\mathbf{w}}^{(r+1)}$
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

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).

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 variavle and texplanatory he 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).

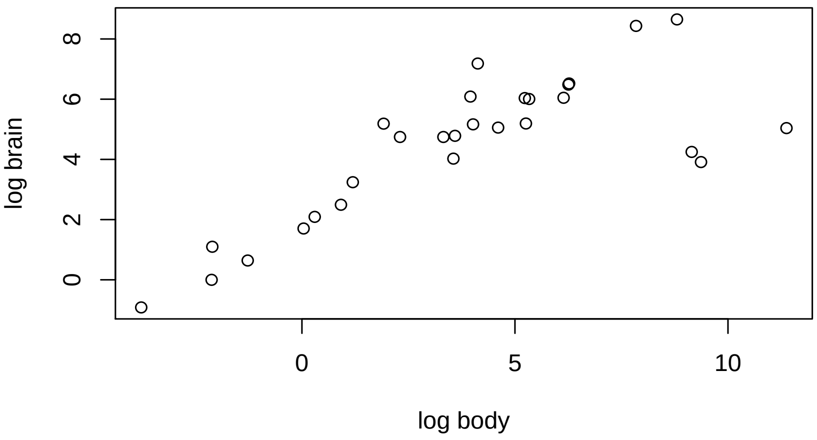

Figure 7.9: A plot of the brain size data.

```
> br.1 <- gamlss(lbrain ~ lbody, data = brains)

GAMLSS-RS iteration 1: Global Deviance = 101.2578
GAMLSS-RS iteration 2: Global Deviance = 101.2578

> 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 ctiteria 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 $0.1, 0.2, \ldots, 1$ 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

$$\hat{\mu} \quad = \quad \begin{cases} -3.072 + 0.750x, & \text{with probability } 0.107 \\ \;\;1.909 + 0.750x, & \text{with probability } 0.751 \\ \;\;3.481 + 0.750x, & \text{with probability } 0.141 \end{cases} \qquad (7.16)$$

and $\hat{\sigma} = 0.391$. [Note that the inrecept 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))
+ }
```

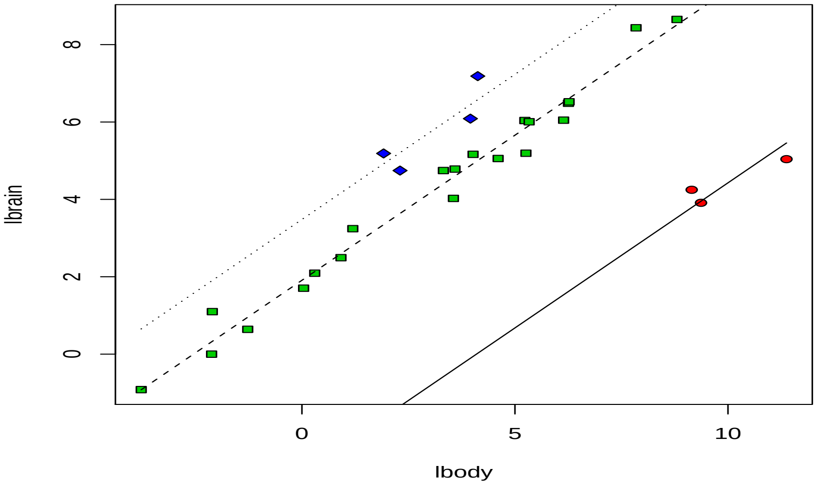

Figure 7.10: A plot of the brain size data together with a plot of the three component fitted means of log brain size (`lbrain`) against log body size (`lbody`), (solid, dashed and dotted for component 1,2 and 3 repsectively).

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
```

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

```
Filliben correlation coefficient  =  0.9962244
******************************************************************
```

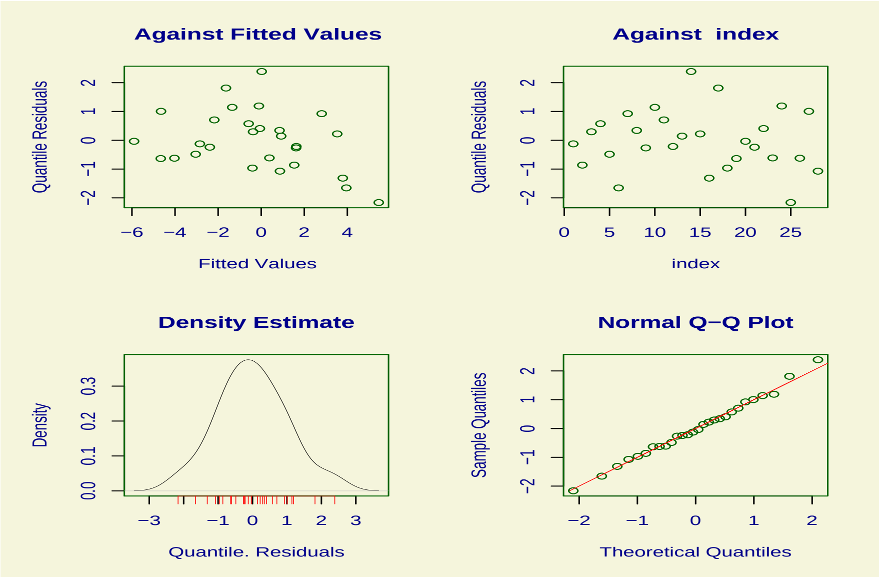

Figure 7.11: The residual plot of model `br.3` for the animal brain size data.

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
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 used in modelling data. Everitt and Hand (1981), Titterington et al. (1985), Lindsay (1995), Böhning (1999) and McLachlan and Peel (2000) are some of the books dedicated exclusively on mixture distributions. Aitkin, Francis and Hinde (2005) include useful chapters related to mixture distributions.

7.9 Exercises

7.9.1 Exercise 1

Here we analyse the acidity data, which records the acidity index for 155 lakes in the Northeastern United States [previously analysed as a mixture of gaussian distributions on the log scale by Crawford *et al.*(1992, 1994)]. These 155 observations are the log acidity indices for the lakes.

- a) Load the acidity data, print the variable name and obtain a histogram of the acidity index values:

  ```
  data(acidity)
  names(acidity)
  hist(y)
  ```

- b) Fit a mixture of two normal distributions (with different means and variances) [note gamlssMXfits fits the mixture from n=20 different starting values and chooses the best mixture fit]:

  ```
      mm<-gamlssMXfits(n=20,y~1, family=NO, K=2)
      mm
  ```

- c) Calculate the probability density function for the fitted mixture and plot it with the histogram of the acidity index values

  ```
  fy<-dMX(y=seq(1,8,.1), mu=list(6.249, 4.33),
             sigma=list(exp(-.6535), exp(-.988)),
              p=list(  0.4039988, 0.5960012) )
      hist(y,freq=FALSE)
      lines(seq(1,8,.1),fy, col="red")
  ```

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 (2.14) 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 (2.16) or (2.18) 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 $k = 1, \ldots, 4$.

When the GAMLSS models $\mathcal{M}_0$ and $\mathcal{M}_1$ contain nonparametric additive terms as in (2.15) and (2.17), 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. GAIC($\sharp$) = GD + $\sharp.df$, 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 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 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 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 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` : sulpher 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 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

object a gamlss object.

scope a formula giving terms which might be dropped or added. For the function dropterm the default is the model formula. For the function addterm the scope 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.

scale scale is not used in gamlss

test it takes value "none" for no test and "Chisq" for a χ^2 test statistic relative to the original model.

k the multiple of the degrees of freedom used for the penalty in the GAIC. Note k = 2 gives the original AIC, k = log(n) gives SBC.

sorted If TRUE the results are sorted in the order of the GAIC from the lowest (the best model) to the highest (the worst model).

trace if 'TRUE' additional information may be given on the fits as they are tried.

... : 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:

```
> 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 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 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 $x6$ 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 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1
```

Among the two way interactions `x4:x5` is highly significant with a p-value of less that 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

`object` a `gamlss` object. This is used as the initial model in the stepwise search

`scope` defines the range of models examined in the stepwise search. For the function `stepGAIC.VR()` this should be either a single formula, or a list containing components `upper` and `lower`, both formulae.

For the function `stepGAIC.CH` the `scope` 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.

`direction` the mode of stepwise search, can be one of `both`, `backward`, or `forward`, with a default of `both` which performs forward stepwise model selection. If the `scope` argument is missing the default for `direction` is `backward`

`trace` if positive, information is printed during the running of `stepGAIC`. Larger values may give more information on the fitting process

`keep` a filter function whose input is a fitted model object and the associated `AIC` statistic, and whose output is arbitrary. Typically `keep` will select a subset of the components of the object and return them. The default is not to keep anything.

`steps` 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.

`scale` scale is not used in `gamlss`

`what` which distribution parameter is being modelled, default `what="mu"`

`k` the multiple of the degrees of freedom used for the penalty in the GAIC criterion. `k = 2` gives the genuine Akaike information criterion (AIC), while `k = log(n)` gives the Schwartz Bayesian criterion (SBC).

... any additional arguments to `extractAIC`. (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))
```

```
Distribution parameter:  mu
Start:  AIC= 319.16
 y ~ 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 ~ 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
```

```
     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

$x1
~1 + x1 + cs(x1)

$x2
~1 + x2 + cs(x2)

$x3
~1 + x3 + cs(x3)

$x4
~1 + x4 + cs(x4)
```

```
$x5
~1 + x5 + cs(x5)

$x6
~1 + x6 + cs(x6)
```

The function `gamlss.scope` has th 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 spine, `cs`

`arg` any additional arguments required by the smoother, (for example `df` for `cs`)

`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
```

```
> mod6 <- stepGAIC(mod5, gs, additive = TRUE)

Distribution parameter:  mu
Start:  y ~ 1; AIC= 353.7146
Trial:  y ~  x1 + 1 + 1 + 1 + 1 + 1; AIC= 338.0354
Trial:  y ~  1 + x2 + 1 + 1 + 1 + 1; AIC= 343.0487
Trial:  y ~  1 + 1 + x3 + 1 + 1 + 1; AIC= 349.2046
Trial:  y ~  1 + 1 + 1 + x4 + 1 + 1; AIC= 355.0206
Trial:  y ~  1 + 1 + 1 + 1 + x5 + 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; AIC= 337.3823
Trial:  y ~  x1 + x2 + 1 + 1 + 1 + 1; AIC= 328.781
Trial:  y ~  x1 + 1 + x3 + 1 + 1 + 1; AIC= 332.942
Trial:  y ~  x1 + 1 + 1 + x4 + 1 + 1; AIC= 339.6497
Trial:  y ~  x1 + 1 + 1 + 1 + x5 + 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; AIC= 328.8904
Trial:  y ~  x1 + cs(x2) + 1 + 1 + 1 + 1; AIC= 333.3071
Trial:  y ~  x1 + x2 + x3 + 1 + 1 + 1; AIC= 325.665
Trial:  y ~  x1 + x2 + 1 + x4 + 1 + 1; AIC= 327.6493
Trial:  y ~  x1 + x2 + 1 + 1 + x5 + 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 $x1$ is selected for inclusion. In the second step, since $x1$ is already in the model, $cs(x1)$ is tried together with all the linear terms from the rest of the variable. In this second step the variable $x2$ 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 `x1` is `1 + x1 + cs(x1)` requiring `1`, (i.e. no $x1$ variable), `x1`, (linear in $x1$) and `cs(x1)`, (smooth term in $x1$), to be tested in sequence. The terms for `x1` in the `gamlss.scope` can only move one step up or down from the current term in `x1` in the sequence. Hence, for example, the model in `x1` can change from `1` only to `x1`, but from `x1` to either `1` or `cs(x1)`, and from `cs(x1)` only to `x1`.

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
+ 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:** LGAclaims in package **gamlss.data** 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.

```
> data(LGAclaims)
> with(LGAclaims, plot(data.frame(Claims, L_Popdensity, L_KI, L_Accidents,
+     L_Population)))
```

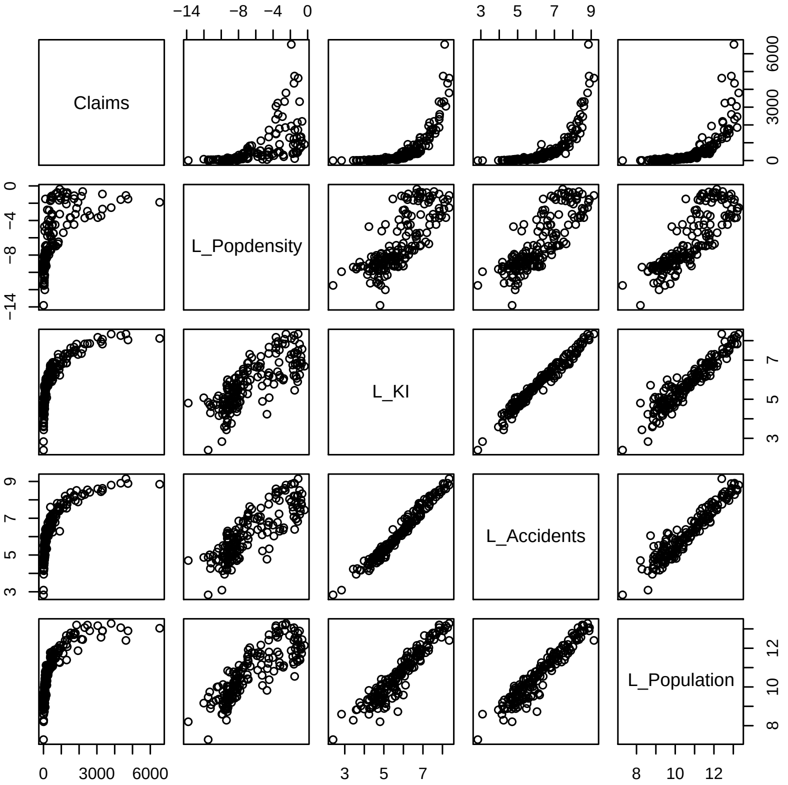

Figure 8.1: The plot of the third party insurance data

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 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 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 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 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
- L_Population 0.9984 1912.5

Step:  AIC= 1899.39
 ~L_Population + L_Accidents

                    Df    AIC
+ L_Popdensity 0.99990 1898.3
```

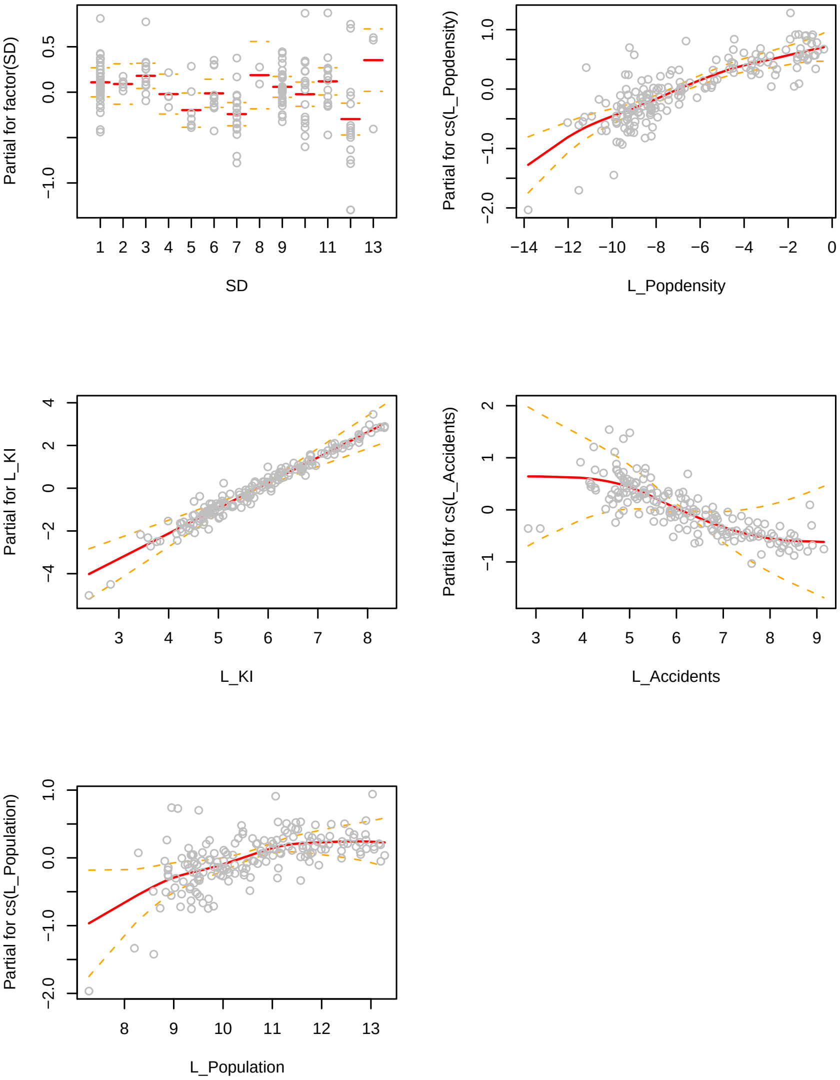

Figure 8.2: The additive parameter plots for the μ model

```
<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 ((quote())) GAMLSS model in which the required hyperparameters are denoted by `p[number]`, e.g. `quote(gamlss(y∼cs(x,df=p[1]),sigma.fo=∼ cs(x,df=p[2]),data=abdom))`

`parameters` the starting parameter values in the search for the optimum hyperparameters and/or non-linear parameters, e.g. `parameters=c(3,3)`

`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. `others=quote(nx<-x^p[3])` where nx is now in the model formula

`penalty` specifies the penalty in the GAIC, (the default is 2) e.g. `penalty=3`

`steps` the steps in the parameter(s) taken during the optimization procedure (see for example the `ndeps` option in the control function for `optim()`), 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 `lower=c(1,1)`. 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 `upper=c(30,10)`. This does not apply if a method other than the default method ”L-BFGS-B” is used `method` the method used in `optim()` 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 `control` argument of the `R` function `optim()`

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 ± 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.]

```
> 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)
```

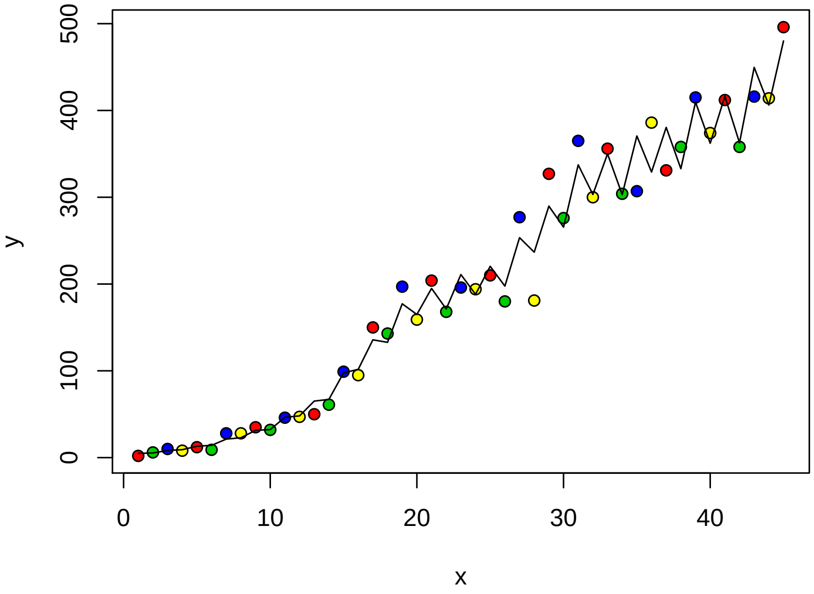

Figure 8.3: AIDS data: data and fitted values from the `aids.1` model

```
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 `mu` (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. 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)
> 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)
```

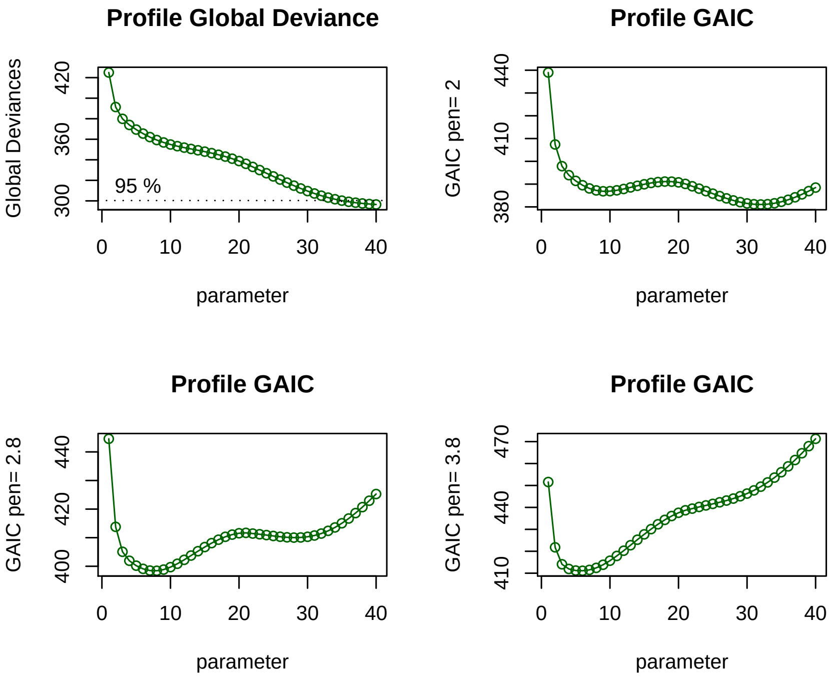

Figure 8.4: Profile global deviance and GAIC for different smoothing degrees of freedom fitted in the NBI model to the AIDS data. (a) global deviance (b) GAIC with penalty $\sharp = 2$, (c) $\sharp = 2.8$ (d) $\sharp = 3.8$, plotted against the smoothing degrees of freedom in the model for μ

```
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 **??**. We model the abdominal circumference, Y, using a BCT(μ, σ, ν, τ) 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 $\boldsymbol{\lambda} = (df_{\mu}, df_{\sigma}, \xi)$. 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
...
...
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
> plot(aids$x, aids$y)
> lines(aids$x, fitted(mod1), col = "red")
> lines(aids$x, fitted(mod2), col = "blue")
```

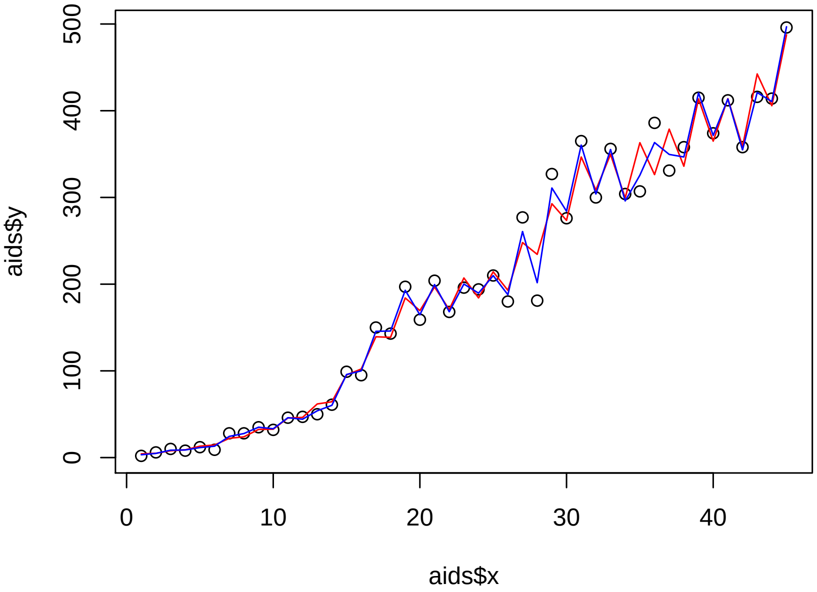

Figure 8.5: AIDS data: fitted NBI models with *df* = 10 (red solid line) and *df* = 30 (blue dashed, line)

```
$par
[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 mu, 1.32 degrees of freedom for smoothing the sigma 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 ploted 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

  ```
  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.

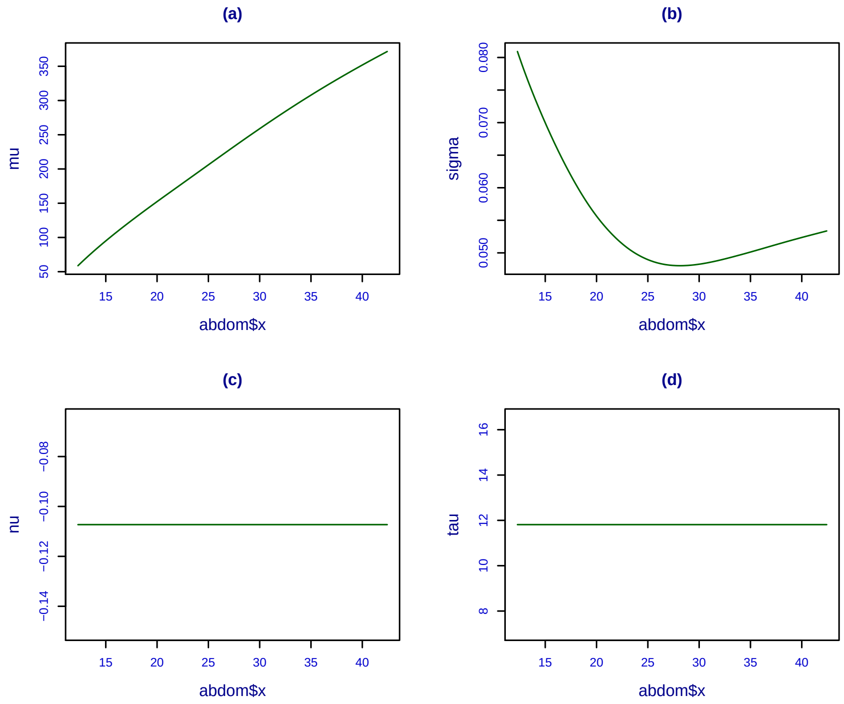

Figure 8.6: Abdominal circumference data: fitted μ, σ, ν and τ against x

```
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

Nonlinear functions in GAMLSS

9.1 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 9.2 describes the use of the function `nlgamlss()` while Section 9.3 describe the `nl()` function.

9.2 The function nlgamss

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.

9.2.1 The data

Data summary:

R **data file:** `la` 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` : unkown

`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 `la`. 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 $\mu = h_1(x) = x\left[1 + \beta_{11}\exp\left(-\beta_{12}/x\right)\right]$ and a constant for the scale parameter σ (and any shape parameters ν and τ). Here we use the reparameterization $\mu = h_1(x) = x\left[1 + (1 - e^{b_{11}})\exp\left(-b_{12}/x\right)\right]$ to improve model fitting. Note that the nonlinear function $\mu = h_1(x)$ passes through the origin, i.e. $\mu = 0$ for x=0.

9.2.2 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 9.1).

```
> 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 mu 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")
```

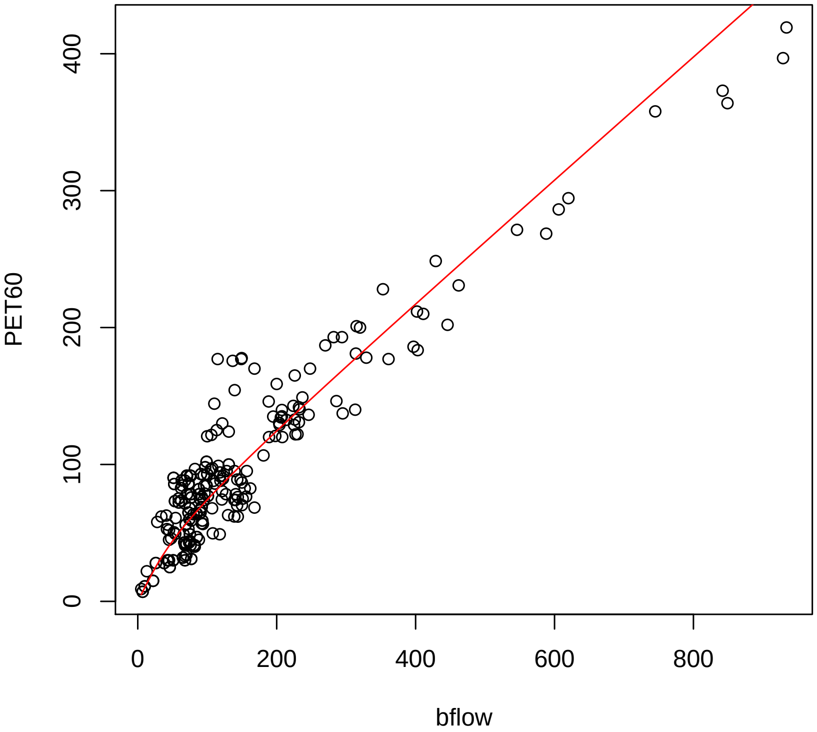

Figure 9.1: Plotting the data together with the fitted values for `mu` for the fitted model `mod1`.

```
+     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 that 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 9.1 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 9.2 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.

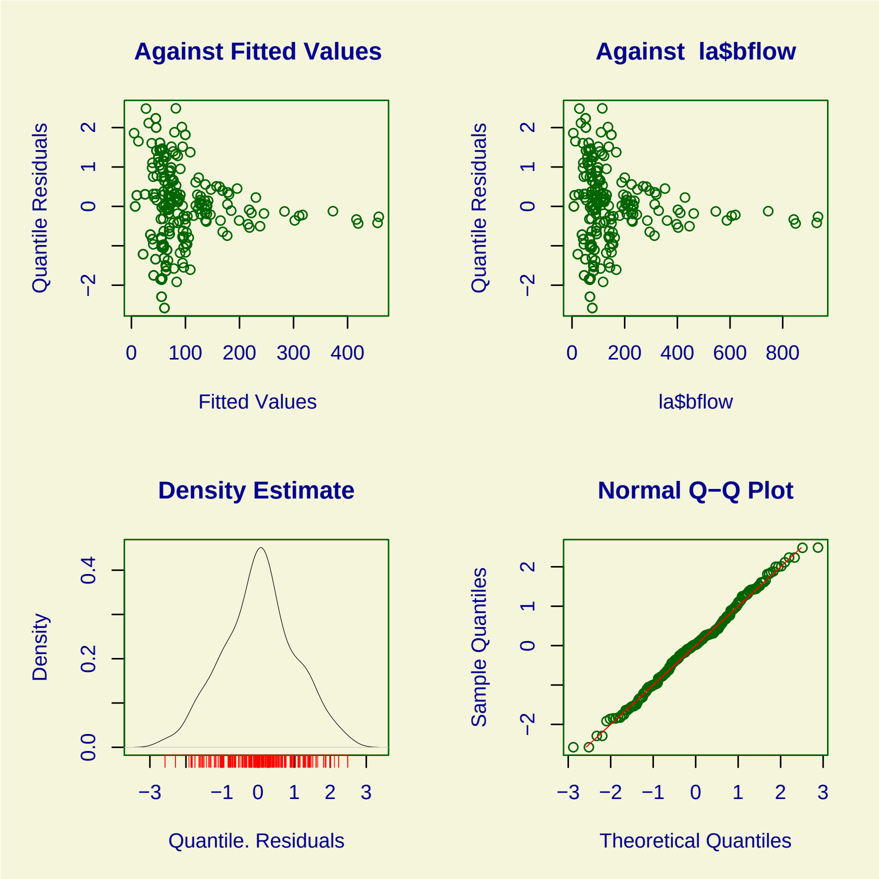

Figure 9.2: The plot of the residuals from model `mod1` .

```
> 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 <- nlgamlss(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:
nlgamlss(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 9.3 shows the residuals for model `mod3` while figure 9.4 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 commads to plot figures 9.3 and 9.4:

```
> 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
```

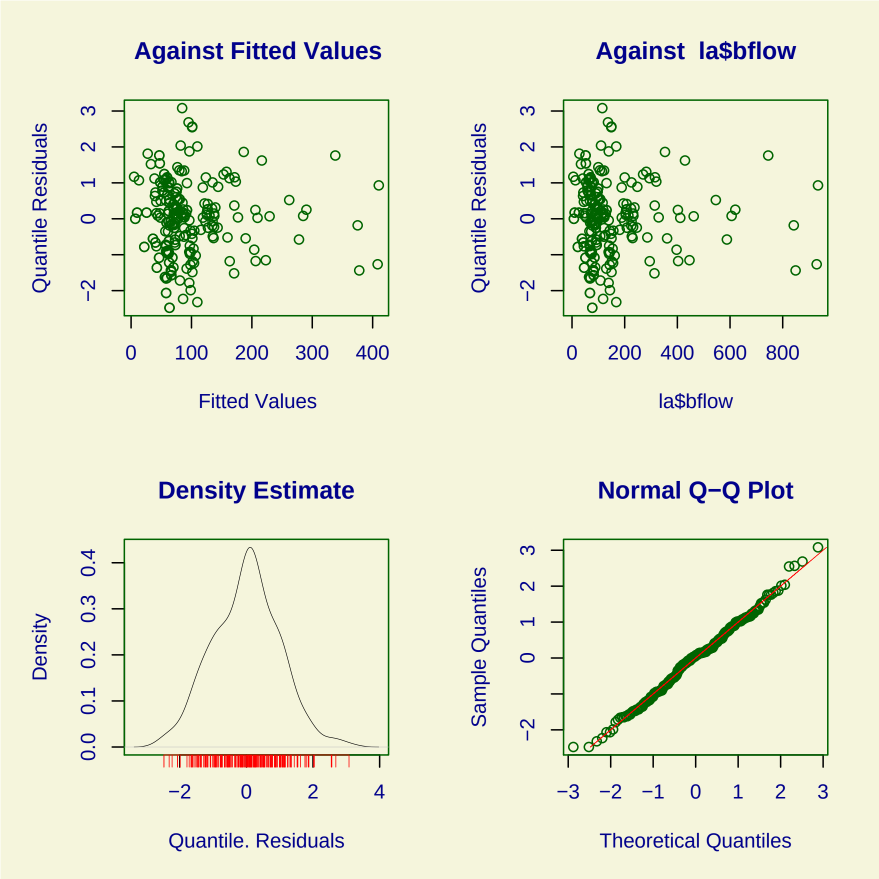

Figure 9.3: The plot of the residuals from model `mod3` using the command `plot(mod3, xvar=la$bflow)`.

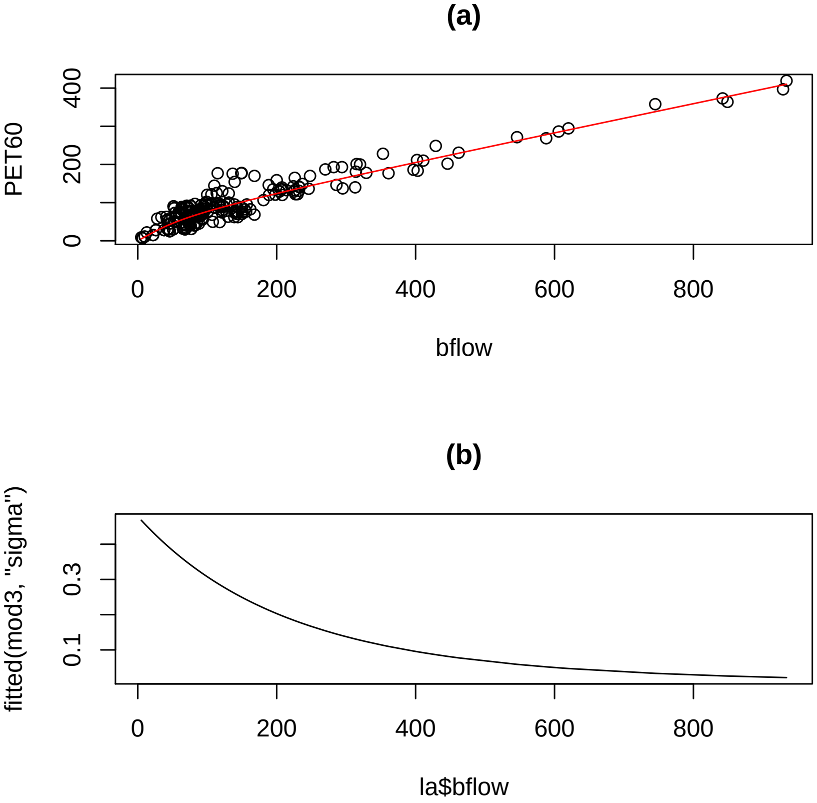

Figure 9.4: Model `mod3` fited values: (a) the data together with the fitted values for μ, (b) the fitted vales for σ

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  1.00000000 -0.15832731 -0.061589927
[5,]  0.11221156  0.01478113 -0.066744373 -0.15832731  1.00000000 -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 $\log\tau$.

9.3 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 9.2.2 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 9.5 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
```

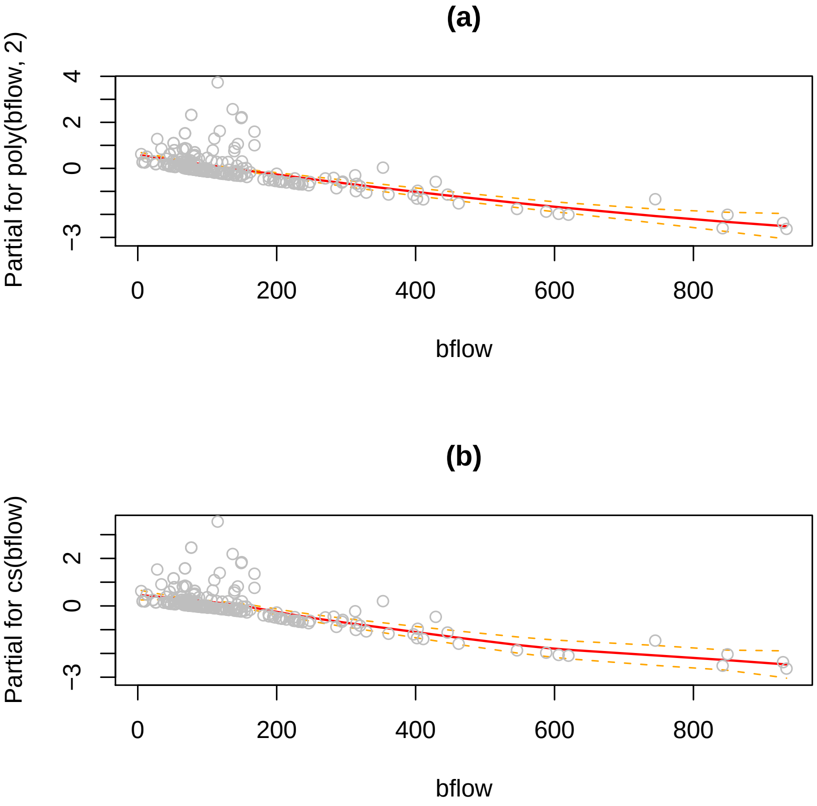

Figure 9.5: Fitted values, Standard errors and partial residuals for the `sigma` parameter model: (a) model `mod4` fitted quadratic polynomials , (b) model `mod5` fitted smoothing cubic spline

```
 Degrees of Freedom for the fit: 9.000726 Residual Deg. of Freedom   241.9993
Global Deviance:     2218.8
            AIC:     2236.8
            SBC:     2268.53
```

9.4 Exercises

- Q1 The table 10.3 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 9.1: 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 10

Centile estimation

10.1 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 10.1. 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 (2.15), as smooth nonparametric functions of x, i.e. $Y \sim BCT(\mu, \sigma, \nu, \tau)$ where

$$\begin{aligned} g_1(\mu) &= h_1(x) \\ g_2(\sigma) &= h_2(x) \\ g_3(\nu) &= h_3(x) \\ g_4(\tau) &= h_4(x) \end{aligned} \tag{10.1}$$

and, for $k = 1, 2, 3, 4$, $g_k(.)$ are known monotonic link functions, and $h_k(x)$ are smooth nonparametric functions of x.

The model selection procedure comprised of choosing link functions $g_k(.)$, 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 $\tau > 0$).

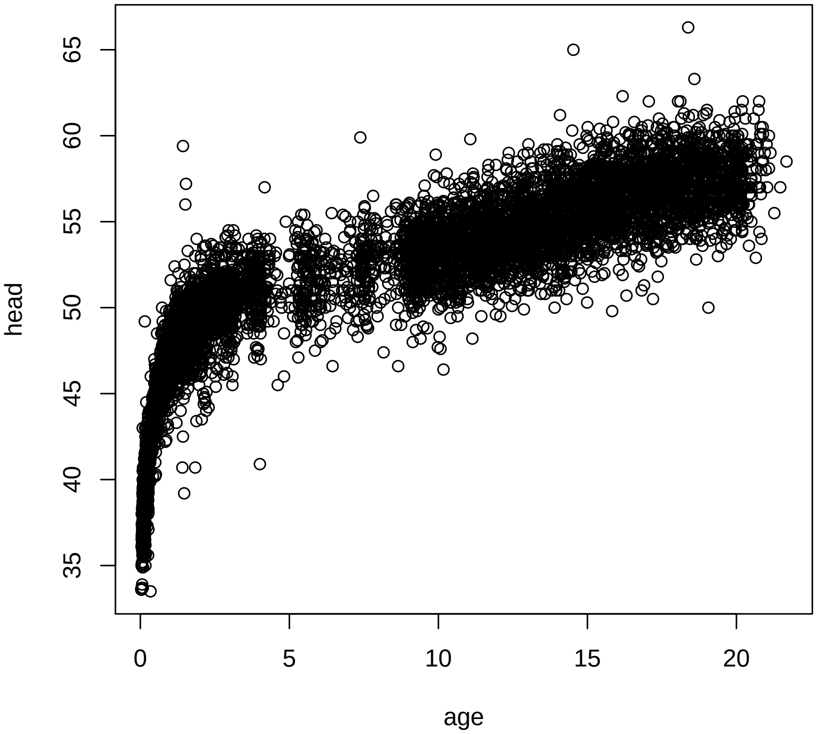

Figure 10.1: Observed Head Circumference of Males (From the Fourth Dutch Growth Study) Against Age.

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 $GAIC(\sharp)$, 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 $\sharp$ in GAIC($\sharp$) are shown in Table 10.1. [Note that in general the $GAIC(\sharp)$ can potentially have multiple local minima (especially for low values of $\sharp$) 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 10.1. Here we show only the $GAIC(\sharp = 2)$ code. The penalty $\sharp$ 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 ($\sharp$) 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 10.1. 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 $\sharp = 3$ (see row 3 of the Table 10.1) 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 10.1: 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 10.1

Table 10.1 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 10.1.

```
> 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 10.2 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 10.2. Of the seven distributions shown, the Box-Cox t (BCT), Johnson's Su (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 10.2 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 10.2 were selected using find.hyper using the same techniques described above.]

The conclusion from Table 10.2 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.

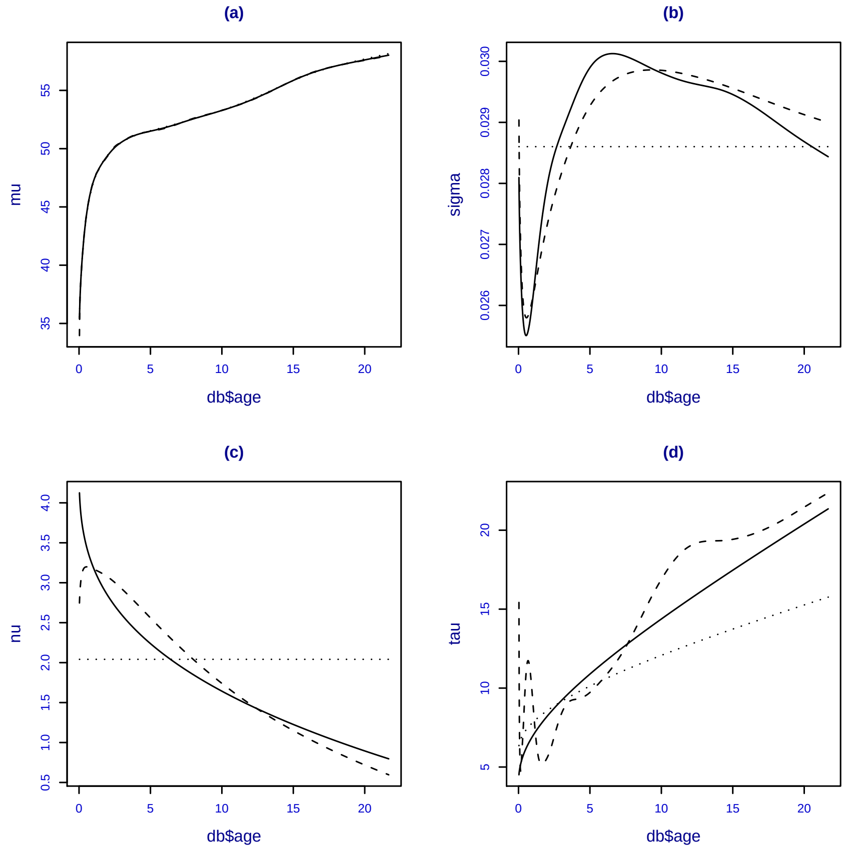

Figure 10.2: The fitted parameters against age for the 'best' BCT models using AIC (---), GAIC(3) (—), and SBC (···), criteria: (a) μ (b) σ (c) ν (d) τ.

Table 10.2: 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 (10.1), for the chosen model $BCT(12.3, 5.7, 2, 2, 0.33)$, are displayed in solid lines in Figure 10.2. 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 10.2(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 10.3 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 10.4 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 10.4. 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 10.4, 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

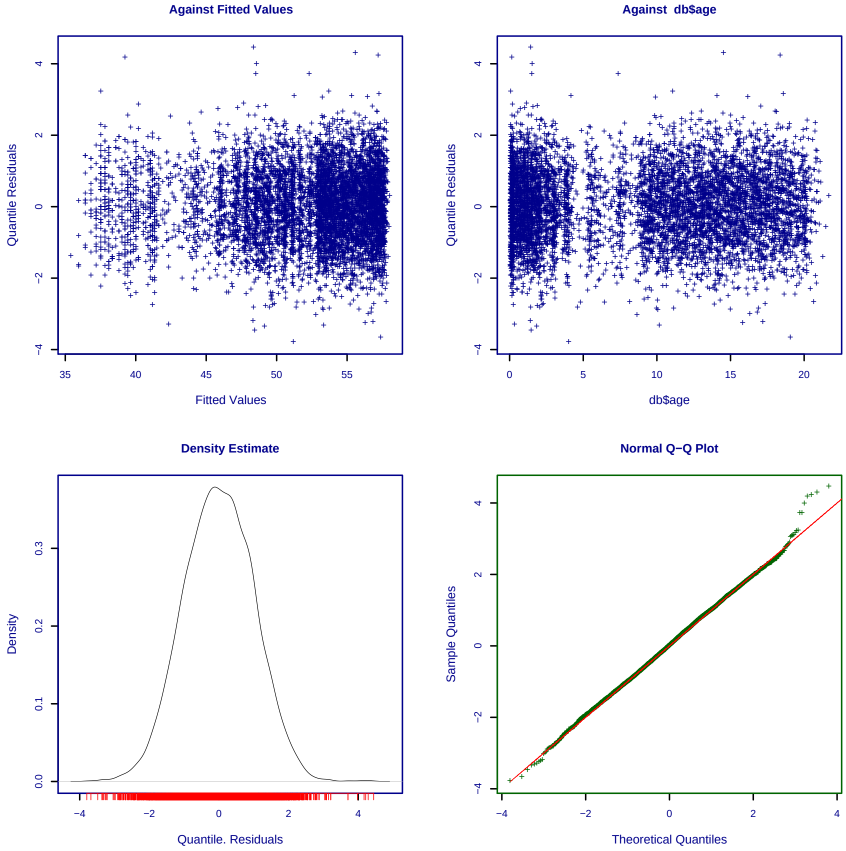

Figure 10.3: The Residuals From Model $BCT(12.3, 5.7, 2, 2, 0.33)$. (a) against fitted values of μ (b) against age (c) kernel density estimate (d) normal QQ plot.

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 10.4 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 10.4). The sixteen age ranges used 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 $\{r_{gi}, i = 1, 2, .., n_i\}$ be the residuals in age group g, with mean $\bar{r}_g$ and standard deviation s_g, for $g = 1, 2, .., G$. The following statistics $\mathsf{Z}_{g1}, \mathsf{Z}_{g2}, \mathsf{Z}_{g3}, \mathsf{Z}_{g4}$ 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 $\mathsf{Z}_{g1} = n_g^{1/2}\bar{r}_g$, $\mathsf{Z}_{g2} = \left\{s_g^{2/3} - [1 - 2/(9n_g - 9)]\right\} / \{2/(9n_g - 9)\}^{1/2}$ 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 $Q_j = \sum_{g=1}^{G} \mathsf{Z}_{gj}^2$ 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

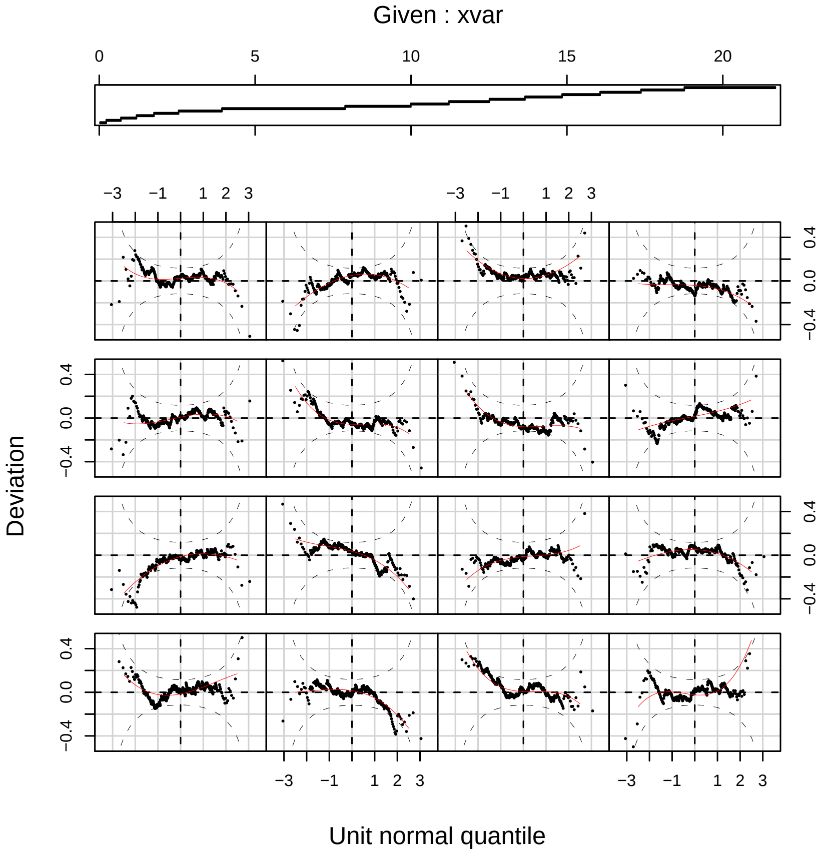

Figure 10.4: Worm Plot of the Residuals From Model $BCT(12.3, 5.7, 2, 2, 0.33)$. The 16 detrended QQ plots, from bottom left to top right plot, correspond to 16 age ranges (displayed in steps above the worm plot from 0 to 22 years).

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 $|\mathsf{Z}_{gj}|$ greater than 2 be considered as indicative of significant inadequacies in the model. Note that significant positive (or negative) values $\mathsf{Z}_{gj} > 2$ (or $\mathsf{Z}_{gj} < 2$) 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 10.5 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
```

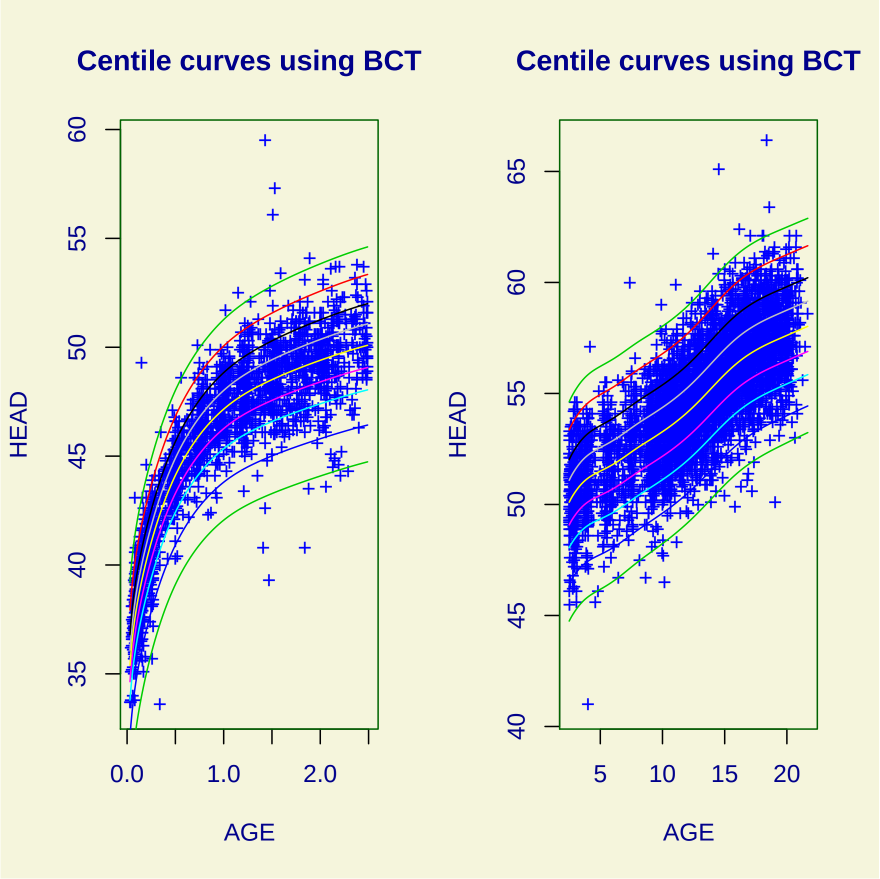

Figure 10.5: Observed Head Circumference With Nine Fitted Model Centile Curves (0.4, 2, 10, 25, 50, 75, 90, 98, 99.6), From Model $BCT(12.3, 5.7, 2, 2, 0.33)$, against Age: (a) 0-2.5 years (b) 2.5-22 years

Figure 10.5 provides nine fitted model centile curves, defined by (10.2) 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

$$y_\alpha = \{ \begin{array}{ll} \mu[1+\sigma\nu t_{\tau,\alpha}]^{1/\nu} & \text{if } \nu \neq 0 \\ \mu \exp[\sigma t_{\tau,\alpha}] & \text{if } \nu = 0, \end{array} \qquad (10.2)$$

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 10.3: 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 10.5(a) and for age range 2.5 to 22 years in Figure 10.5(b), for clarity of presentation. Table 10.3 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 10.2 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.]

10.2 Exercises

10.2.1 Exercise 1

- 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 `dbbmi` and contain the variables `bmi` and `age`.

  (a) Plot `bmi` against `age`.

  (b) Transform age to a new variable `nage <- dbbmi$ageô.377` and plot `bmi` against `nage`.

  (c) Fit a BCPE distribution to `bmi` using a P-splines in `nage` i.e. `pb(nage)` for the predictors for parameter μ. How many degrees of freedom were used for the smoothing? (Use the function `edf()` or `m1$mu.df`).

  (d) Use the fitted values from (c) as starting values for fitting a BCPE distribution to bmi using P-splines in `nage` for the predictors for parameters μ, σ, ν and τ. What are the effective degrees of freedom for all the parameters? [You can use the function `edfAll()`].

  (e) Plot the fitted parameters for the fitted model in (c) using `fitted.plot()`.

  (f) Obtain a centile plot for the fitted model in (d) using `centiles()` or `centiles.split()`.

  (g) Investigate the residuals from the fitted model in (c) using e.g. `plot()`, `wp()` (worm plot) and `Q.stats()` (Q-statistics).

Comment: Numerical optimization could be used to select the power parameter ξ applied to transform the explanatory variable age. Here we applied the transformation power $\xi = 0.377$ found to be appropriate if an GAIC(3) is used [see Rigby and Stasinopoulos (2004)]. The code used for this optimization is:

```
nage<-dbbmi$age^3
m3<-gamlss(bmi~pb(nage), sigma.fo=~pb(nage), nu.fo=~pb(nage),
          tau.fo=~pb(nage),data=dbbmi, family=BCPE)
fn<-function(p)
{
nage<- dbbmi$age^p
cat(p,"\n")
AIC(m4<-gamlss(bmi~pb(nage), sigma.fo=~pb(nage), nu.fo=~pb(nage),
     tau.fo=~pb(nage), data=dbbmi, family=BCPE, start.from=m3), k=3)
}
optim(0.3, fn, method= "L-BFGS-B")
```

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 **NO**(μ,σ), 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 Gausian) 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 **NO**(μ,σ), is

$$f_Y(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \, \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right] \tag{A.1}$$

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 $Var(Y) = \sigma^2$, 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

$$f(y|\mu,\sigma,\nu) = \frac{1}{\sqrt{2\pi}\sigma|\mu|^{\nu/2}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2|\mu|^\nu}\right] \tag{A.2}$$

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

$$f_Y(y|\mu,\sigma) = \frac{1}{\sigma}\left\{\exp\left[-\left(\frac{y-\mu}{\sigma}\right)\right]\right\}\left\{1+\exp\left[-\left(\frac{y-\mu}{\sigma}\right)\right]\right\}^{-2} \tag{A.3}$$

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

$$f_Y(y|\mu,\sigma) = \frac{1}{\sigma}\ \exp\left[\left(\frac{y-\mu}{\sigma}\right) - \exp\left(\frac{y-\mu}{\sigma}\right)\right] \tag{A.4}$$

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 **RG**(μ,σ) is defined by

$$f_Y(y|\mu,\sigma) = \frac{1}{\sigma}\,\exp\left\{-\left(\frac{y-\mu}{\sigma}\right) - \exp\left[-\frac{(y-\mu)}{\sigma}\right]\right\} \tag{A.5}$$

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 **exGAUS**(μ,σ), is defined as

$$f_Y(y|\mu,\sigma,\nu) = \frac{1}{\nu}\exp\left[\frac{\mu-y}{\nu} + \frac{\sigma^2}{2\nu^2}\right]\Phi\left(\frac{y-\mu}{\sigma} - \frac{\sigma}{\nu}\right) \tag{A.6}$$

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 **PE**(μ,σ,ν), is defined by

$$f_Y(y|\mu,\sigma,\nu) \;=\; \frac{\nu\exp[-\left|\frac{z}{c}\right|^{\nu}]}{2c\sigma\Gamma\left(\frac{1}{\nu}\right)} \tag{A.7}$$

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 $\nu \to \infty$.

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 $f_S(s) = s^{1/\nu}\exp(-s)/\Gamma\left(\frac{1}{\nu}\right)$.

Second parameterization (PE2)

An alternative parameterization, the power exponential type 2 distribution, denoted by **PE2**(μ,σ,ν), is defined by

$$f_Y(y|\mu,\sigma,\nu) \;=\; \frac{\nu\exp[-|z|^{\nu}]}{2\sigma\Gamma\left(\frac{1}{\nu}\right)} \tag{A.8}$$

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 **TF**(μ,σ,ν), is defined by

$$f_Y(y|\mu,\sigma,\nu) = \frac{1}{\sigma B\left(\frac{1}{2},\frac{\nu}{2}\right)\nu^{\frac{1}{2}}}\left[1 + \frac{(y-\mu)^2}{\sigma^2\nu}\right]^{-\frac{\nu+1}{2}} \tag{A.9}$$

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

$$f_Y(y|\mu,\sigma,\nu,\tau) = e^{\nu z}\{|\sigma|\, B(\nu,\tau)\,[1+e^z]^{\nu+\tau}\}^{-1} \tag{A.10}$$

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 $Var(Y) = \sigma^2\left[\Psi^{(1)}(\nu) + \Psi^{(1)}(\tau)\right]$, from McDonald (1996), p437.

A.3.2 Generalized t distribution (GT)

This pdf of the generalized t distribution, denoted by **GT**(μ,σ,ν,τ), is defined by

$$f_Y(y|\mu,\sigma\,\nu,\tau) = \tau\left\{2\sigma\nu^{1/\tau}B\left(1/\tau,\nu\right)[1+|z|^\tau/\nu]^{\nu+(1/\tau)}\right\}^{-1} \tag{A.11}$$

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 $\mathbf{JSUo}(\mu,\sigma,\nu,\tau)$, is defined by

$$f_Y(y|\mu,\sigma\,\nu,\tau) = \frac{\tau}{\sigma}\frac{1}{(r^2+1)^{\frac{1}{2}}}\frac{1}{\sqrt{2\pi}}\ \exp\left[-\frac{1}{2}z^2\right] \tag{A.12}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$ and $\tau > 0$, and where

$$z = \nu + \tau\sinh^{-1}(r) = \nu + \tau\log\left[r + (r^2+1)^{\frac{1}{2}}\right], \tag{A.13}$$

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 $Var(Y) = \sigma^2\frac{1}{2}(\omega-1)\left[\omega\cosh(2\nu/\tau)+1\right]$, 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 $\mathbf{JSU}(\mu,\sigma,\nu,\tau)$, is defined by

$$f_Y(y|\mu,\sigma\,\nu,\tau) == \frac{\tau}{c\sigma}\frac{1}{(r^2+1)^{\frac{1}{2}}}\frac{1}{\sqrt{2\pi}}\ \exp\left[-\frac{1}{2}z^2\right] \tag{A.14}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$, $\tau > 0$, and where

$$z = -\nu + \tau\sinh^{-1}(r) = -\nu + \tau\log\left[r + (r^2+1)^{\frac{1}{2}})\right], \tag{A.15}$$

$$r = \frac{y - (\mu + c\sigma w^{\frac{1}{2}}\sinh\Omega)}{c\sigma},$$

$$c = \left\{\frac{1}{2}(w-1)\left[w\cosh(2\Omega)+1\right]\right\}^{-\frac{1}{2}},$$

$w = \exp(1/\tau^2)$ 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 $\mathbf{NET}(\mu,\sigma,\nu,\tau)$, 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

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{c}{\sigma}\begin{cases} \exp\left\{-\frac{z^2}{2}\right\}, & \text{when} \quad |z| \le \nu \\ \exp\left\{-\nu|z| + \frac{\nu^2}{2}\right\}, & \text{when} \quad \nu < |z| \le \tau \\ \exp\left\{-\nu\tau\log\left(\frac{|z|}{\tau}\right) - \nu\tau + \frac{\nu^2}{2}\right\}, & \text{when} \quad |z| > \tau \end{cases} \tag{A.16}$$

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

$$f_Y(y|\mu,\sigma\,\nu,\tau) = \frac{c}{\sqrt{2\pi}\sigma(1+r^2)^{1/2}}e^{-z^2/2} \tag{A.17}$$

where

$$z = \frac{1}{2}\left\{\exp\left[\tau\sinh^{-1}(r)\right] - \exp\left[-\nu\sinh^{-1}(r)\right]\right\}$$

and

$$c = \frac{1}{2}\left\{\tau\exp\left[\tau\sinh^{-1}(r)\right] + \nu\exp\left[-\nu\sinh^{-1}(r)\right]\right\}$$

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 $u = r + \left(r^2+1\right)^{1/2}$. Hence $z = \frac{1}{2}\left(u^\tau - u^{-\nu}\right)$. 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

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{2}{\sigma}\,f_{Z_1}(z)\,F_{Z_1}(\nu z) \tag{A.18}$$

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

[1] since NET involves the Student-t distribution with ($\nu\tau$-1) degrees of freedom

2 distribution with $f_{Z_1}(z) = \alpha^{-1} \exp\left[-|z|^\tau/\tau\right]$, 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 $Var(Y) = \sigma^2 V(Z) = \sigma^2 \left\{E(Z^2) - [E(Z)]^2\right\}$ where $Z = (Y - \mu)/\sigma$ and $E(Z) = \text{sign}(\nu)\tau^{1/\tau}\left[\Gamma\left(\frac{2}{\tau}\right)/\Gamma\left(\frac{1}{\tau}\right)\right] pBEo\left(\frac{\nu^\tau}{1+\nu^\tau}, \frac{1}{\tau}, \frac{2}{\tau}\right)$, and $E(Z^2) = \tau^{2/\tau}\Gamma\left(\frac{3}{\tau}\right)/\Gamma\left(\frac{1}{\tau}\right)$, 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 $E(Y) = \mu + \sigma\text{sign}(\nu)\left\{2\nu^2/\left[\pi(1+\nu^2)\right]\right\}^{1/2}$ and $Var(Y) = \sigma^2\left\{1 - 2\nu^2/\left[\pi(1+\nu^2)\right]\right\}$, 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

$$f_Y(y|\mu, \sigma\,\nu, \tau) = \frac{2}{\sigma} f_{Z_1}(z)\;\Phi(\omega) \tag{A.19}$$

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

$$E(Z) = \frac{2\tau^{1/\tau}\nu}{\sqrt{\pi}\Gamma\left(\frac{1}{\tau}\right)(1+\nu^2)^{(2/\tau)+(1/2)}} \sum_{n=0}^{\infty} \frac{\Gamma\left(\frac{2}{\tau} + n + \frac{1}{2}\right)}{(2n+1)!!}\left(\frac{2\nu^2}{1+\nu^2}\right)^n \tag{A.20}$$

and $E(Z^2) = \tau^{1/\tau}\Gamma\left(\frac{3}{\tau}\right)/\Gamma\left(\frac{1}{\tau}\right)$, 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

$$f_Y(y|\mu, \sigma\,\nu, \tau) = \frac{c}{\sigma}\left\{\exp\left[-\frac{1}{2}|\nu z|^\tau\right]\;I(y < \mu) + \exp\left[-\frac{1}{2}\left|\frac{z}{\nu}\right|^\tau\right]\;I(y \geq \mu)\right\} \tag{A.21}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = \nu\tau/\left[(1+\nu^2)2^{1/\tau}\Gamma\left(\frac{1}{\tau}\right)\right]$, 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 $E(Z) = 2^{1/\tau}\Gamma\left(\frac{2}{\tau}\right)\left(\nu - \frac{1}{\nu}\right)/\Gamma\left(\frac{1}{\tau}\right)$ and $E(Z^2) = 2^{2/\tau}\Gamma\left(\frac{3}{\tau}\right)\left(\nu^3 + \frac{1}{\nu^3}\right) / \left[\Gamma\left(\frac{1}{\tau}\right)\left(\nu + \frac{1}{\nu}\right)\right]$, 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

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{c}{\sigma}\left\{\exp[-|z|^{\nu}]\ I(y<\mu) + \exp[-|z|^{\tau}]\ I(y \geq \mu)\right\} \tag{A.22}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$, and $\tau > 0$, and where $z = (y-\mu)/\sigma$ and $c = \left[\Gamma\left(1+\frac{1}{\nu}\right) + \Gamma\left(1+\frac{1}{\tau}\right)\right]^{-1}$, 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 $E(Z) = c\left[\frac{1}{\tau}\Gamma\left(\frac{2}{\tau}\right) - \frac{1}{\nu}\Gamma\left(\frac{2}{\nu}\right)\right]$ and $E(Z^2) = c\left[\frac{1}{\nu}\Gamma\left(\frac{3}{\nu}\right) + \frac{1}{\tau}\Gamma\left(\frac{3}{\tau}\right)\right]$.

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

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{2}{\sigma}\ f_{Z_1}(z)\ F_{Z_1}(\nu z) \tag{A.23}$$

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

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{2}{\sigma}\ f_{Z_1}(z)\ F_{Z_2}(w) \tag{A.24}$$

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_1} 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 $E(Z) = \nu\tau^{1/2}\Gamma\left(\frac{\tau-1}{2}\right) / \left[\pi^{1/2}(1+\nu^2)^{1/2}\Gamma\left(\frac{\tau}{2}\right)\right]$ 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 $ST3(\mu,\sigma,\nu,\tau)$, defined by

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{c}{\sigma}\ \left\{1 + \frac{z^2}{\tau}\left[\nu^2\ I(y<\mu) + \frac{1}{\nu^2}\ I(y \geq \mu)\right]\right\} \tag{A.25}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = 2\nu / \left[\sigma\left(1+\nu^2\right) B\left(\frac{1}{2}, \frac{\tau}{2}\right)\ \tau^{1/2}\right]$, 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 $E(Z) = 2\tau^{1/2}(\nu^2 - 1)/\left[(\tau - 1)B\left(\frac{1}{2}, \frac{\tau}{2}\right)\nu\right]$ and $E(Z^2) = \tau\left(\nu^3 + \frac{1}{\nu^3}\right)/\left[(\tau - 2)\left(\nu + \frac{1}{\nu}\right)\right]$, 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

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma}\left\{\left[1 + \frac{z^2}{\nu}\right]^{-(\nu+1)/2} I(y < \mu) + \left[1 + \frac{z^2}{\tau}\right]^{-(\tau+1)/2} I(y \geq \mu)\right\} \tag{A.26}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = 2\left[\nu^{1/2} B\left(\frac{1}{2}, \frac{\nu}{2}\right) + \tau^{1/2} B\left(\frac{1}{2}, \frac{\tau}{2}\right)\right]^{-1}$.

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 $E(Z^2) = \frac{c}{2}\left\{\left[\tau^{3/2} B\left(\frac{1}{2}, \frac{\tau}{2}\right)/(\tau - 2)\right] + \left[\nu^{3/2} B\left(\frac{1}{2}, \frac{\nu}{2}\right)/(\nu - 2)\right]\right\}$, 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 **ST5**(μ,σ,ν,τ), 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 $c = \left[2^{a+b-1}(a + b)^{1/2} B(a, b)\right]^{-1}$ and $\nu = (a - b)/\left[ab(a + b)\right]^{1/2}$ and $\tau = 2/(a + b)$.

Here $E(Y) = \mu + \sigma E(Z)$ where $E(Z) = (a - b)(a + b)^{1/2}\Gamma\left(a - \frac{1}{2}\right)\Gamma\left(a - \frac{1}{2}\right)/\left[2\Gamma(a)\Gamma(b)\right]$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z^2) = (a + b)\left[(a - b)^2 + a + b - 2\right]/\left[4(a - 1)(b - 1)\right]$, 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 **EXP**(μ), is defined by

$$f_Y(y|\mu) = \frac{1}{\mu}\exp\left\{-\frac{y}{\mu}\right\} \tag{A.27}$$

for $y > 0$, where $\mu > 0$ and where $E(Y) = \mu$ and $Var(Y) = \mu^2$.

A.5 Continuous two parameter distribution in $\Re^+$

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

$$f_Y(y|\mu,\sigma) = \frac{1}{(\sigma^2\mu)^{1/\sigma^2}} \frac{y^{\frac{1}{\sigma^2}-1}\, e^{-y/(\sigma^2\mu)}}{\Gamma(1/\sigma^2)} \tag{A.28}$$

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

$$f_Y(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y} \exp\left\{-\frac{[\log(y)-\mu]^2}{2\sigma^2}\right\} \tag{A.29}$$

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 $Z = (Y^\nu - 1)/\nu(\textbf{if} \ \ \nu \neq 0) + \log(Y)(\textbf{if} \ \ \nu = 0)$. 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

$$f_Y(y|\mu,\sigma,\nu) = \frac{y^{\nu-1}}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right] \tag{A.30}$$

for $y > 0$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$, and where $z = (y^\nu - 1)/\nu(\textbf{if}\ \nu \neq 0) + \log(y)(\textbf{if}\ \nu = 0)$. 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

$$f_Y(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2 y^3}} \exp\left[-\frac{1}{2\mu^2\sigma^2 y}(y-\mu)^2\right] \tag{A.31}$$

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

$$f_Y(y|\mu,\sigma) = \frac{\sigma y^{\sigma-1}}{\mu^\sigma} \exp\left[-\left(\frac{y}{\mu}\right)^\sigma\right] \tag{A.32}$$

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 $E(Y) = \mu\,\Gamma\left(\frac{1}{\sigma}+1\right)$ and $Var(Y) = \mu^2\left\{\Gamma\left(\frac{2}{\sigma}+1\right) - \left[\Gamma\left(\frac{1}{\sigma}+1\right)\right]^2\right\}$, 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

$$f_Y(y|\mu,\sigma) = \sigma\mu y^{\sigma-1} e^{-\mu y^\sigma} \tag{A.33}$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$, Johnson *et al.* (1994), p686. The mean of Y in this parameterization (A.33) is $E(Y) = \mu^{-1/\sigma}\,\Gamma\left(\frac{1}{\sigma}+1\right)$ and the variance of Y is $Var(Y) = \mu^{-2/\sigma}\left\{\Gamma\left(\frac{2}{\sigma}+1\right) - \left[\Gamma\left(\frac{1}{\sigma}+1\right)\right]^2\right\}$.

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

$$f_Y(y|\mu,\sigma) = \frac{\sigma}{\beta}\left(\frac{y}{\beta}\right)^{\sigma-1} \exp\left\{-\left(\frac{y}{\beta}\right)^\sigma\right\} \tag{A.34}$$

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 $Var(Y) = \mu^2\left\{\Gamma(\frac{2}{\sigma}+1)\left[\Gamma(\frac{1}{\sigma}+1)\right]^{-2} - 1\right\}$.

A.6 Continuous three parameter distribution in $\Re^+$

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

$$Z \quad = \quad \begin{cases} \frac{1}{\sigma\nu}\left[\left(\frac{Y}{\mu}\right)^{\nu} - 1\right], & \text{if} \quad \nu \neq 0 \\ \frac{1}{\sigma}\log(\frac{Y}{\mu}), & \text{if} \quad \nu = 0 \end{cases} \tag{A.35}$$

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

$$f_Y(y) = \frac{y^{\nu-1}\exp(-\frac{1}{2}z^2)}{\mu^{\nu}\sigma\sqrt{2\pi}\Phi(\frac{1}{\sigma|\nu|})} \tag{A.36}$$

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 Lopatatzidis and Green (2000), and is defined as

$$f_Y(y|\mu,\sigma,\nu) = \frac{|\nu|\theta^{\theta}z^{\theta}\exp\{-\theta z\}}{\Gamma(\theta)y} \tag{A.37}$$

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 $E(Y) = \mu\Gamma\left(\theta + \frac{1}{\nu}\right) / \left[\theta^{1/\nu}\Gamma(\theta)\right]$ and $Var(Y) = \mu^2\left\{\Gamma(\theta)\Gamma\left(\theta+\frac{2}{\nu}\right) - \left[\Gamma\left(\theta+\frac{1}{\nu}\right)\right]^2\right\} / \left\{\theta^{2/\nu}\left[\Gamma(\theta)\right]^2\right\}$. 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

$$f_Y(y|\mu,\sigma,\nu) = \frac{|\mu|y^{\mu\nu-1}}{\Gamma(\nu)\sigma^{\mu\nu}}\exp\left\{-\left(\frac{y}{\sigma}\right)^{\mu}\right\} \tag{A.38}$$

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

The mean and variance of $Y \sim$ **GG2**(μ,σ,ν) can be obtained from those of **GG**(μ,σ,ν) since **GG**(μ,σ,ν) $\equiv$ **GG2**(ν,$\mu\theta^{-1/\nu}$,θ) and **GG2**(μ,σ,ν) $\equiv$ **GG**($\sigma\nu^{1/\mu}$,$\left[\mu^2\nu\right]^{-1/2}$, μ).

A.6.3 Generalized inverse Gaussian distribution (GIG)

The parameterization of the generalized inverse Gaussian distribution, denoted by **GIG**(μ,σ,ν), is defined as

$$f_Y(y|\mu,\sigma,\nu) = \left(\frac{c}{\mu}\right)^{\nu} \left[\frac{y^{\nu-1}}{2K_\nu\left(\frac{1}{\sigma^2}\right)}\right] \exp\left[-\frac{1}{2\sigma^2}\left(\frac{cy}{\mu}+\frac{\mu}{cy}\right)\right] \qquad \text{(A.39)}$$

for $y > 0$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$, where $c = \left[K_{\nu+1}\left(1/\sigma^2\right)\right]\left[K_\nu\left(1/\sigma^2\right)\right]^{-1}$ 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 $Var(Y) = \mu^2\left[2\sigma^2(\nu+1)/c + 1/c^2 - 1\right]$. **GIG**($\mu$,$\sigma$,$\nu$) is a reparameterization of the generalized inverse Gaussian distribution of Jorgensen (1982) . Note also that **GIG**(μ,σ,-0.5) $\equiv$ **IG**(μ, $\sigma\mu^{-1/2}$) a reparameterization of the inverse Gaussian distribution.

A.6.4 Zero adjusted Gamma distribution (ZAGA)

The zero adjusted Gamma distribution is appropriate when the response variable Y takes values from zero to infinity including zero, i.e. [0,∞). Hence $Y = 0$ has non zero probability ν. The pdf of the zero adjusted Gamma distribution, denoted by **ZAGA**(μ,σ,ν), is defined by

$$f_Y(y|\mu,\sigma,\nu) = \begin{cases} \nu & \text{if } y = 0 \\ (1-\nu)\left[\frac{1}{(\sigma^2\mu)^{1/\sigma^2}} \frac{y^{\frac{1}{\sigma^2}-1}\, e^{-y/(\sigma^2\mu)}}{\Gamma(1/\sigma^2)}\right] & \text{if } y > 0 \end{cases} \qquad \text{(A.40)}$$

for $0 \le y < \infty$, where $0 < \nu < 1$, $\mu > 0$ and $\sigma > 0$ with $E(Y) = (1-\nu)\mu$ and $Var(Y) = (1-\nu)\mu^2(\nu+\sigma^2)$.

A.6.5 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,∞). Hence $Y = 0$ has non zero probability ν. The pdf of the zero adjusted inverse Gaussian distribution, denoted by **ZAIG**(μ,σ,ν), is defined by

$$f_Y(y|\mu,\sigma,\nu) = \begin{cases} \nu & \text{if } y = 0 \\ (1-\nu)\frac{1}{\sqrt{2\pi\sigma^2 y^3}} \exp\left[-\frac{1}{2\mu^2\sigma^2 y}(y-\mu)^2\right] & \text{if } y > 0 \end{cases} \qquad \text{(A.41)}$$

for $0 \le y < \infty$, 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 $\Re^+$

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 **BCT**(μ,σ,ν,τ), 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

$$f_Y(y|\mu,\sigma,\nu,\tau) = \frac{y^{\nu-1} f_T(z)}{\mu^{\nu} \sigma F_T(\frac{1}{\sigma|\nu|})} \tag{A.42}$$

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.42), 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

$$\begin{aligned} f_Y(y|\mu,\sigma,\nu,\tau) &= |\sigma| y^{\sigma\nu-1} \left\{ \mu^{\sigma\nu} \, B(\nu,\tau) \, [1+(y/\mu)^{\sigma}]^{\nu+\tau} \right\}^{-1} \\ &= \frac{\Gamma(\nu+\tau)}{\Gamma(\nu)\Gamma(\tau)} \frac{\sigma (y/\mu)^{\sigma\nu}}{y\,[1+(y/\mu)^{\sigma}]^{\nu+\tau}} \end{aligned} \tag{A.43}$$

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 $E(Y) = \mu B\left(\nu + \frac{1}{\sigma}, \tau - \frac{1}{\sigma}\right)/B\left(\nu,\tau\right)$ for $-\nu < \frac{1}{\sigma} < \tau$ and $E(Y^2) = \mu^2 B\left(\nu + \frac{2}{\sigma}, \tau - \frac{2}{\sigma}\right)/B\left(\nu,\tau\right)$ for $-\nu < \frac{2}{\sigma} < \tau$, McDonald (1996), p434. Note the by setting $\nu = 1$ in A.43 we obtain the *Burr* distribution:

$$f_Y(y|\mu,\sigma,\tau) = \frac{\tau\sigma(y/\mu)^{\sigma}}{y\,[1+(y/\mu)^{\sigma}]^{\tau+1}}. \tag{A.44}$$

By setting $\sigma = 1$ in A.43 we obtain the *Generalized Pareto* distribution:

$$f_Y(y|\mu,\nu,\tau) = \frac{\Gamma(\nu+\tau)}{\Gamma(\nu)\Gamma(\tau)} \frac{(\mu^{\tau} y^{\nu-1}}{(y+\mu)^{\nu+\tau}}. \tag{A.45}$$

A.8 Continuous two parameter distribution in $\Re[0,1]$

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 **BE**(μ,σ), is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{B(\alpha, \beta)}\, y^{\alpha-1}(1-y)^{\beta-1} \tag{A.46}$$

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

$$f_Y(y|\mu, \sigma, \nu, \tau) = \begin{cases} p_0 & \text{if } y = 0 \\ (1-p_0-p_1)\frac{1}{B(\alpha,\beta)}\, y^{\alpha-1}(1-y)^{\beta-1} & \text{if } 0 < y < 1 \\ p_1 & \text{if } y = 1 \end{cases} \tag{A.47}$$

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 $GB1(\mu, \sigma, \nu, \tau)$, is given by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{\tau\nu^\beta y^{\tau\alpha-1}(1-y^\tau)^{\beta-1}}{B(\alpha, \beta)[\nu + (1-\nu)y^\tau]^{\alpha+\beta}} \tag{A.48}$$

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, $GB1(\mu, \sigma, \nu, \tau)$ 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 $BE(\mu, \sigma)$ distribution is a special case of $GB1(\mu, \sigma, \nu, \tau)$ 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 $y = 0, 1, 2, ..., n$, 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

$$p_Y(y|\mu,\sigma) = \frac{\Gamma(n+1)}{\Gamma(y+1)\Gamma(n-y+1)} \frac{\Gamma(\frac{1}{\sigma})\Gamma(y+\frac{\mu}{\sigma})\Gamma[n+\frac{(1-\mu)}{\sigma}-y]}{\Gamma(n+\frac{1}{\sigma})\Gamma(\frac{\mu}{\sigma})\Gamma(\frac{1-\mu}{\sigma})} \tag{A.49}$$

for $y = 0, 1, 2, \ldots, n$, 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 **BI**(n,μ) distribution is the limiting distribution of **BB**(n,μ,σ) as $\sigma \to 0$. For $\mu = 0.5$ and $\sigma = 0.5$, **BB**(n,μ,σ) 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 **PO**(μ), is given by

$$p_Y(y|\mu) = P(Y=y|\mu) \quad = \quad \frac{e^{-\mu}\mu^y}{y!} \tag{A.50}$$

where $y = 0, 1, 2, \ldots$, 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 **NBI**(μ,σ), 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 $y = 0, 1, 2, ...,$ 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 **NBII**(μ,σ), 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 $y = 0, 1, 2, ...,,$ 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 $y = 0, 1, 2, ..., \infty$ 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

$$p_Y(y|\mu,\sigma,\nu) = \frac{e^{-\mu\nu}}{\Gamma(1/\sigma)}\left[1+\mu\sigma(1-\nu)\right]^{-1/\sigma} S \qquad \text{(A.51)}$$

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 $y = 0, 1, 2, ..., \infty$ 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

$$p_Y(y|\mu,\sigma,\nu) = \frac{\mu^y K_{y+\nu}(\alpha)}{(\alpha\sigma)^{y+\nu} y! K_\nu(\frac{1}{\sigma})} \tag{A.52}$$

where $\alpha^2 = \frac{1}{\sigma^2} + \frac{2\mu}{\sigma}$, for $y = 0, 1, 2, ..., \infty$ 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

$$p_Y(y|\mu,\sigma,\nu) = \frac{(\mu/c)^y K_{y+\nu}(\alpha)}{y!\,(\alpha\sigma)^{y+\nu}\,K_\nu\left(\frac{1}{\sigma}\right)} \tag{A.53}$$

for $y = 0, 1, 2, ..., \infty$, where $\alpha^2 = \sigma^{-2} + 2\mu(c\sigma)^{-1}$. The mean of Y is given by $E(Y) = \mu$ and the variance by $Var(Y) = \mu + \mu^2\left[2\sigma(\nu+1)/c + 1/c^2 - 1\right]$.

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

$$p_Y(y|\mu,\sigma) \quad = \quad \begin{cases} \sigma + (1-\sigma)e^{-\mu}, & \text{if} \quad y = 0 \\ (1-\sigma)\frac{\mu^y}{y!}e^{-\mu}, & \text{if} \quad y = 1, 2, 3, \ldots \end{cases} \tag{A.54}$$

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

$$p_Y(y|\mu,\sigma) \quad = \quad \begin{cases} \sigma + (1-\sigma)e^{-\left(\frac{\mu}{1-\sigma}\right)}, & \text{if} \quad y = 0 \\ (1-\sigma)\frac{\mu^y}{y!(1-\sigma)^y}e^{-\left(\frac{\mu}{1-\sigma}\right)}, & \text{if} \quad y = 1, 2, 3, \ldots \end{cases} \tag{A.55}$$

The mean of Y in (A.55) 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] **Akaike, H.** (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**: 716–723.

[4] **Akaike, H.** (1983). Information measures and model selection. *Bulletin of the International Statistical Institute*, **50**: 277–290.

[5] **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.

[6] **Ambler, G.** (1999). *`fracpoly()`: Fractional Polynomial Model*. `S-PLUS`.

[7] **Anscombe, F. J.** (1950). Sampling theory of the negative binomial and logarithmic series approximations. *Biometrika*, **37**: 358–382.

[8] **Azzalini, A.** (1986). Further results on a class of distributions which includes the normal ones. *Statistica*, **46**: 199:208.

[9] **Azzalini, A. and Bowman, A. W.** (1990). A look at some data on the Old Faithful geyser. *Applied Statistics*, **39**: 357–365.

[10] **Azzalini, A. and Capitanio, A.** (1985). A class of distributions which includes the normal ones. *Scand. J. Statist.*, **12**: 171–178.

[11] **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.

[12] **Besag, J. and Higdon, P.** (1999). Bayesian analysis of agriculture field experiments (with discussion). *Journal of the Royal Statistical Society, B*, **61**: 691–746.

[13] **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.

[14] **Box, G. E. P. and Cox, D. R.** (1964). An analysis of transformations (with discussion). *J. R. Statist. Soc. B.*, **26**: 211–252.

[15] **Box, G. E. P. and Tiao, G. C.** (1973). *Bayesian Inference in Statistical Analysis*. Wiley, New York.

[16] **Carroll, R. J. and Ruppert, D.** (1982). Robust estimation in heteroscedastic linear models. *Ann. Statist.*, **10**: 429–441.

[17] **Chambers, J. M. and Hastie, T. J.** (1992). *Statistical Models in S*. Chapman & Hall, London.

[18] **Chappas, C. and Corina-Borja, M.** (2006). A stylometric analysis of newspapers, periodicals and news scripts. *Journal of Quantitative Linguistics*, **13**: 285–312.

[19] **Claeskens, G. and Hjort, N. L.** (2003). The focused information criterion. *J. Am. Statist. Ass.*, **98**: 900–916.

[20] **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.

[21] **Cole, T. J. and Green, P. J.** (1992). Smoothing reference centile curves: the LMS method and penalized likelihood. *Statist. Med.*, **11**: 1305–1319.

[22] **Consul, P. C.** (1989). *Generalized Poisson Distributions*. Marcel Dekker, New York.

[23] **Crisp, A. and Burridge, J.** (1994). A note on nonregular likelihood functions in heteroscedastic regression models. *Biometrika*, **81**: 585–587.

[24] **Crowder, M. J., Kimber, A. C., Smith R. L. and Sweeting, T. J.** (1991). *Statistical Analysis of Reliability Data*. Chapman and Hall, London.

[25] **CYTEL Software Corporation** (2001). *EGRET for Windows*. CYTEL Software Corporation, Cambridge, Massachusetts.

[26] **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.

[27] **Davidian, M. and Carroll, R. J.** (1988). A note on extended quasi-likelihood. *J. R. Statist. Soc.*, **50**: 74–82.

[28] **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.

[29] **de Boor, C.** (1978). *A Practical Guide to Splines*. Springer, New York.

[30] **de Jong, P. and Heller, G. Z.** (2007). *Generalized Linear Models for Insurance Data*. Cambridge University Press.

[31] **Dean, C., Lawless, J. F. and Willmot, G. E.** (1989). A mixed Poisson-inverse-Gaussian regression model. *Canadian Journal of Statistics*, **17**: 171–181.

[32] **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.

[33] **DiCiccio, T. J. and Monti, A. C.** (2004). Inferential Aspects of the Skew Exponential Power Distribution. *J. Am. Statist. Ass.*, **99**: 439–450.

[34] **Draper, D.** (1995). Assessment and propagation of model uncertainty (with discussion). *J. R. Statist. Soc. B.*, **57**: 45–97.

[35] **Dunn, P. K. and Smyth, G. K.** (1996). Randomised quantile residuals. *J. Comput. Graph. Statist.*, **5**: 236–244.

[36] **Efron, B.** (1986). Double exponential families and their use in generalized linear regression. *J. Am. Statist. Ass.*, **81**: 709–721.

[37] **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.

[38] **Evans, D. A.** (1953). Experimental evidence concerning contagious distributions in ecology. *Biometrika*, **40**: 186–211.

[39] **Fernandez, C. and Steel, M. F. J.** (1998). On Bayesian Modelling of Fat Tails and Skewness. *J. Am. Statist. Ass.*, **93**: 359–371.

[40] **Fernandez, C., Osiewalski, J. and Steel, M. J. F.** (1995). Modeling and inference with v-spherical distributions. *J. Am. Statist. Ass.*, **90**: 1331–1340.

[41] **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.

[42] **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.

[43] **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.

[44] **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.

[45] **Gilchrist, R.** (2000). Regression models for data with a non-zero probability of a zero response. *Commun. Statist. Theory Meth.*, **29**: 1987–2003.

[46] **Green, P. J. and Silverman, B. W.** (1994). *Nonparametric Regression and Generalized Linear Models*. Chapman and Hall, London.

[47] **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.

[48] **Hansen, B.** (1994). Autoregressive conditional density estimation. *International Ecomomic Review*, **35**: 705–730.

[49] **Hastie, T.** (2006). **gam***: Generalized Additive Models*. R package version 0.98.

[50] **Hastie, T. J. and Tibshirani, R. J.** (1990). *Generalized Additive Models*. Chapman and Hall, London.

[51] **Hastie, T. J. and Tibshirani, R. J.** (1993). Varying coefficient models (with discussion). *J. R. Statist. Soc. B.*, **55**: 757–796.

[52] **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.

[53] **Hjort, N. L. and Claeskens, G.** (2003). Frequentist model average estimation. *J. Am. Statist. Ass.*, **98**: 879–899.

[54] **Ihaka, R. and Gentleman, R.** (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, **5**: 299–314.

[55] **Johnson, N. L.** (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, **36**: 149–176.

[56] **Johnson, N. L., Kotz, S. and Balakrishnan, N.** (1994). *Continuous Univariate Distributions, Volume I, 2nd edn.* Wiley, New York.

[57] **Johnson, N. L., Kotz, S. and Balakrishnan, N.** (1995). *Continuous Univariate Distributions, Volume II, 2nd edn.* Wiley, New York.

[58] **Johnson, N. L., Kotz, S. and Kemp, A. W.** (2005). *Univariate Discrete Distributions, 3nd edn.* Wiley, New York.

[59] **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.

[60] **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.

[61] **Jørgensen, B.** (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution, Lecture Notes in Statistics No.***9**. Springer-Verlag, New York.

[62] **Karlis, D. and Xekalaki, E.** (2008). The Polygonal Distribution. In: Arnold, B. C., Balakrishnan, N, Minués, M and Sarabia, J. M. (eds.), *Mathematical and Statistical Modeling in Honor of Enrique Castillo*. Birkhauser: Boston.

[63] **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.

[64] **Lambert, D.** (1992). Zero-inflated Poisson Regression with an application to defects in Manufacturing. *Technometrics*, **34**: 1–14.

[65] **Lambert, P. and Lindsey, J.** (1999). Analysing financial returns using regression models based on non-symmetric stable distributions. *Applied Statistics*, **48**: 409–424.

[66] **Lange, K.** (1999). *Numerical Analysis for Statisticians*. Springer, New York.

[67] **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.

[68] **Lee, Y. and Nelder, J. A.** (1996). Hierarchical generalized linear models (with discussion). *J. R. Statist. Soc. B.*, **58**: 619–678.

[69] **Lee, Y. and Nelder, J. A.** (2006). Doudle Hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**: 139–185.

[70] **Lindsey, J.** (1999). On the use of correction for overdispersion. *Applied Statistics*, **48**: 553–561.

[71] **Lindsey, J. K.** (1999). *Models for Repeated Measurments (Second Edition).* Oxford University Press, Oxford.

[72] **Lopatatzidis, A. and Green, P. J.** (2000). Nonparametric quantile regression using the gamma distribution. *submitted for publication.*

[73] **Madigan, D. and Raftery, A. E.** (1994). Model selection and accounting for model uncertainly in graphical models using Occam's window. *J. Am. Statist. Ass.*, **89**: 1535–1546.

[74] **Marx, B.** (2003). *`ps()`: P-Spline Code for GAM's and Univariate GLM Smoothing.* `S-PLUS`.

[75] **McCullagh, P. and Nelder, J. A.** (1989). *Generalized Linear Models, 2nd edn.* Chapman and Hall, London.

[76] **McDonald, J. B.** (1991). Parametric models for partially adaptive estimation with skewed and leptokurtic residuals. *Economic Letters*, **37**: 273–278.

[77] **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.

[78] **McDonald, J. B. and Newey, W. K.** (1988). Partially adaptive estimation of regression models via the generalized t distribution. *Econometric Theory*, **4**: 428–457.

[79] **McDonald, J. B. and Xu, Y. J.** (1995). A generalisation of the beta distribution with applications. *Journal of Econometrics*, **66**: 133–152.

[80] **Nandi, A. K. and Mämpel, D.** (1995). An expension of the generalized Gaussian distribution to include asymmetry. *J. Franklin Inst.*, **332**: 67–75.

[81] **Nelder, J. A. and Lee, Y.** (1992). Likelihood, quasi-likelihood and psuedolikelihood: some comparisons. *J. R. Statist. Soc. B.*, **54**: 273–284.

[82] **Nelder, J. A. and Pregibon, D.** (1987). An extended quasi-likelihood function. *Biometrika*, **74**: 221–232.

[83] **Nelder, J. A. and Wedderburn, R. W. M.** (1972). Generalized linear models. *J. R. Statist. Soc. A.*, **135**: 370–384.

[84] **Nelson, D. B.** (1991). Conditional heteroskedasticity in asset returns: a new approach. *Econometrica*, **59**: 347–370.

[85] **Ortega, J. M. and Rheinboldt, W. C.** (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York.

[86] **P., W. M. and Jones, M. C.** (1999). *Kernel Smoothing*. Chapman & Hall, Essen, Germany.

[87] **Parzen, E.** (1984). Nonparametric statistical data modelling. *Journal of the American Statistical Association*, **74**: 105–131.

[88] **Perks, W. F.** (1932). On some experiments in the graduation of mortality statistics. *Journal of the Institute of Actuaries*, **58**: 12–57.

[89] **Pham-Gia, T. and Duong, Q. P.** (1989). The generalized beta and F distributions in statistical modelling. *Mathematical and Computer Modelling*, **13**: 1613–1625.

[90] **Quesenberry, C. and Hales, C.** (1980). Concentration bands for uniformily plots. *Journal of Statistical Computation and Simulation*, **11**: 41–53.

[91] **Raftery, A. E.** (1996). Approximate Bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika*, **83**: 251–266.

[92] **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.

[93] **Reinsch, C.** (1967). Smoothing by spline functions. *Numerische Mathematik*, **10**: 177–183.

[94] **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.

[95] **Rigby, R. A. and Stasinopoulos, D. M.** (1996a). A semi-parametric additive model for variance heterogeneity. *Statist. Comput.*, **6**: 57–65.

[96] **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.

[97] **Rigby, R. A. and Stasinopoulos, D. M.** (2000). Construction of reference centiles using mean and dispersion additive models. *Statistician*, **49**: 41–50.

[98] **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.

[99] **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.

[100] **Rigby, R. A. and Stasinopoulos, D. M.** (2005). Generalized additive models for location, scale and shape, (with discussion). *Appl. Statist.*, **54**: 507–554.

[101] **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.

[102] **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**.

[103] **Royston, P. and Altman, D. G.** (1994). Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling (with discussion). *Appl. Statist.*, **43**: 429–467.

[104] **Royston, P. and Wright, E. M.** (2000). Goodness-of-fit statistics for age-specific reference intervals. *Statistics in Medicine*, **19**: 2943–2962.

[105] **SAS Institute Inc.** (2000). *Enterprise Miner Software, Version 4*. SAS Institute Inc, Cary, North Carolina.

[106] **Schwarz, G.** (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**: 461–464.

[107] **Sheather, S. J. and Jones, M. C.** (1991). A reliable data-based bandwidth selection mathod for kernel density estimation. *J. R. Statist. Soc. B.*, **53**: 683–690.

[108] **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.

[109] **Silverman, B. W.** (1988). *Density Estimation for Statitsics and Data Analysis*. Chapman & Hall.

[110] **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.

[111] **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.

[112] **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.

[113] **Stasinopoulos, D. M., Rigby, R. A. and Fahrmeir, L.** (2000). Modelling rental guide data using mean and dispersion additive models. *Statistician*, **49**: 479–493.

[114] **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.

[115] **Subbotin, M. T.** (1923). On the law of frequency of errors. *Mathematicheskii Sbornik*, **31**: 296–301.

[116] **Tamura, R. N. and Young, S. S.** (1987). A stabilized moment estimator for the beta binomial distribution. *Biometrics*, **43**: 813–824.

[117] **Theodossiou, P.** (1998). Financial data and the skewed generalized t distribution. *Management Science*, **44**: 1650–1661.

[118] **Tierney, L. and Kadane, J. B.** (1986). Accurate approximations for posterior moments and marginal densities. *J. Am. Statist. Ass.*, **81**: 82–86.

[119] **van Buuren, S. and Fredriks, M.** (2001). Worm plot: a simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, **20**: 1259–1277.

[120] **Venables, W. N. and Ripley, B. D.** (2000). *S Programming*. Springer. ISBN 0-387-98966-8.

[121] **Venables, W. N. and Ripley, B. D.** (2002). *Modern Applied Statistics with S. Fourth Edition*. Springer. ISBN 0-387-98825-4.

[122] **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.

[123] **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.

[124] **Wedderburn, R. W. M.** (1974). Quasi-likelihood functions, generalised linear models and the Gauss-Newton method. *Biometrika*, **61**: 439–447.

[125] **Wilkinson, G. N. and Rogers, C. E.** (1973). Symbolic description of factorial models for analysis of variance. *Appl. Statist.*, **22**: 392–399.

[126] **Wimmer, G. and Altmann, G.** (1999). *Thesaurus of univariate discrete probability distributions*. Stamm Verlag, Essen, Germany.

[127] **Winkelmann, R.** (1997). *Ecometric Analysis of Count Data*. Springer Verlag, Berlin.

[128] **Wood, S. N.** (2001). mgcv: GAMs and Generalised Ridge Regression for R. *R News*, **1**: 20–25.

Index