Aspects of Modeling and Application of Survival-type Data

By

Lida Fallah, M.Sc.

A thesis submitted in partial-fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Biostatistics

Supervisor: Professor John Hinde

School of Mathematics, Statistics and Applied Mathematics,
National University of Ireland, Galway

Submitted February 2018

To my loving parents

To my mom for her devotion and endless support.

To my dad, passed away while I was studying away from home, for his heart warming support. His selflessness will always be remembered and admired.

Contents

List of Figures

List of Tables

Declaration

I declare that this thesis is my own work and that it has not been submitted, in whole or in part, elsewhere.

Lida Fallah

Abstract

Survival analysis is collection of methods for analyzing data where the outcome of interest is the time to an event and some of the observations are censored.

Survival data can arise naturally from studies on, machines' time to break down (also known as reliability) to agricultural experiments on how some environmental conditions affect flowering, to medical cohort studies following-up cancer patients' survival and their reaction to treatments, etc.

In industrial applications there are obvious benefits of progressive censoring (briefly speaking, removing live individuals progressively over time according to a censoring plan) in machine testing where effort, resource, and cost can be saved by early censoring. Furthermore, in agricultural applications, such as the serious threat of certain pests to sugar cane during the planting season or the maturation phase of the cane, biological control assays are used to study the survival of pests under exposure to pesticides. In addition, in recent decades, detecting the associations between patients gene expression profiles and phenotypic data is of increasing interest to aid in improving diagnosis and prognosis of patients and in facilitating treatment discoveries.

To appreciate different aspects of survival data and its applications, this thesis puts together different methods for modeling such data and deals with the unique difficulties that each type of data brings to bear on data analysis.

Acknowledgements

I would like to express my boundless appreciation and gratefulness to my supervisor, Professor John Hinde, for his kind unending and patient instructions and precious support all through the project. Professor John Hinde, I am indebted to you and words would never be enough for expressing this deep gratefulness. It was an honor to work with you.

I should for sure express my thanks to supporters, the Irish Research council and the College of Science for paving the way of my researches by funding me through the postgraduate research scholarship.

I extend my sincere thanks to colleagues at the Insect Pathology Laboratory in Brazil, for providing the insects survival dataset and also researchers in TCGA who made the data on ovarian carcinoma publicly available.

I am grateful to Dr. Emma Holian and Professor Cathal Seoighe for their helpful comments and advice in the annual GRC meetings.

My profound gratitude to the panel of Examiners, Dr. Jochen Einbeck and Professor John Newell, for their constructive comments, suggestions and critiques.

I thank also my friends in Galway who were supportive to me and made joyous time for me. A big thank you goes to Paul Korir for his kind and helpful advice in my first year in NUI Galway.

And last but not least, I owe an ocean depth of heart full gratitude to my loving and caring parents and siblings, particularly to my mom who has always supported and motivated me like an angel and my dad who planted the worth of education and effort in my soul all the way through my life.

Chapter 1

Introduction

Survival analysis has a wide range of important applications in many different areas. The main features of a survival trait are that it is the time until an event occurs, and frequently some of the observations are censored, and essential consideration in survival analysis methodology. In most clinical studies, censoring, in various forms, is a serious issue and must be dealt with appropriately. In reliability experiments, it usually arises whenever the experimenter is unable to observe (either intentionally or unintentionally) the failure times of all individuals under observation.

We start this chapter with some basic concepts and definitions on survival models/data.

Specific distinguishing features of survival data include the following:

- Time to vent data is positive. This means that we confine attention to random variables with positive support (non-negative random variables).

- The most commonly used statistical distributions that fit well to survival data include: Exponential, Weibull, Gamma, Normal and Log-normal distributions.

- Time to event data are usually censored. In survival/reliability data analysis, we observe and record the lifetime of patients or time to failure of products. There could be cases in which, for example, a product has a high reliability and hence obtaining failure data could be a time-consuming task. Furthermore, even if the data can be obtained over a short time period, continuing until failure of even one more individual could be expensive. In regard to the above, in survival data analysis we usually face incomplete data and this is exactly what distinguishes survival analysis from other aspects of statistics.

- Fitting survival models to specifically censored data generally requires using numerical computational methods.

As can be seen from the features of survival data, the distribution of the failure-time random variable plays an important role in assessing the survival of individuals. In the next section, we introduce important functions in survival analysis which are related to failure-time distributions. Here the random variable T indicates a non-negative continuous random variable describing the failure time of individuals under study.

1.1 Characterisations of failure-time distributions

The probability distribution of a continuous failure time T can be given by the cumulative distribution function, quantile function, density function, survival function or the hazard function. Although they all offer a full specification of the distribution of T usually one of them may be more appropriate, or useful, depending on the conditions under which an experiment is undertaken. These functions are defined as follows:

- *Cumulative Distribution Function*: The cumulative distribution function of T, $F(t) = P(T \leq t)$, expresses the probability of failure/death of an individual before time t.

- *Density Function*: The density function for a continuous random variable T is defined as $f(t) = \dfrac{d\,F(t)}{d\,t}$.

- *Quantile Function*: The quantile function t_p is the inverse of the cumulative distribution function and gives the time by which a specific proportion p of individuals fails will have failed, $t_p = F^{-1}(p)$.

- *Survival Function*: Survival function is the complement of the cumulative distribution function and gives the probability of surviving beyond time t.

$$S(t) = P(T > t) = 1 - F(t) \tag{1.1}$$

The survival function is usually a decreasing continuous function such that

$$S(\infty) = \lim_{t\to\infty} S(t) = 0, \quad S(0) = 1.$$

- *Hazard Function*: The hazard function, which is also called the hazard rate function or instantaneous rate of death, is defined as

$$h(t) = \lim_{\Delta t\to 0} \frac{P(t \le T \le t + \Delta t | T \ge t)}{\Delta t} \tag{1.2}$$

The hazard function explains the tendency to fail of individuals over a short time interval, provided that they have not failed prior to that time t. Therefore, for small Δt, we have

$$h(t)\Delta t \approx P(t \le T \le t + \Delta t | T \ge t).$$

The concept of failure/death rate is of paramount importance in life-time studies and is described as the number of failures/deaths in unit time. The failure rate is usually depend upon time. For example, the failure rate of a machine in its fifth year of life is typically much higher than that in its first year.

The hazard function is related to the survival function through the following relations:

$$h(t) = \frac{-S'(t)}{S(t)} = -\frac{d}{dt} \ln \left\{ S(t) \right\} \tag{1.3}$$

$$S(t) = e^{-\int_0^t h(u)\, du} \tag{1.4}$$

In practice, the hazard function is typically of one of five forms: constant, increasing, decreasing, unimodal and bathtub shape. However, in this thesis we are not explicitly interested in the shape of hazard function.

1.2 Aspects of the thesis

Inference based on censored sampling schemes has been studied during the past seven decades for a wide range of lifetime distributions. Survival models and the associated estimation procedures provide a flexible means of modeling survival data. In this thesis we provide insights on different types of survival data and how they can be analyzed when integrated with other types of data.

Figure 1.1 is a schematic diagram of survival-type data, showing a number of fields that survival data can be incorporated and methods for dealing with analysis of data in complicated settings.

In this thesis, Chapter 2 presents preliminaries and background information about different types of data, from censored data, mixture data, clustered longitudinal data, to genetic data arising form microarray experiments and how they fit into the context of survival analysis.

In Chapter 3, mixture data which are subject to censoring from heterogeneous populations are modeled and methods for estimation of parameters are discussed. The effect of different types of censoring schemes, for example Type-II censoring and progressive censoring are explored through conducting two simulation studies.

Chapter 4 is devoted to the analysis of discrete survival data when the data arise from a longitudinal study measuring the survival times over discrete time intervals. The data used for this chapter were collected by the Insect Pathology Laboratory of ESALQ-USP, Piracicaba, São Paulo, Brazil, with the aim of reducing the damage by the termite *Heterotermes tenuis* to sugarcane in Brazil.

In Chapter 5, methods for analyzing data from clinical trials and microarray experiments are introduced and ways of incorporating genetic data and clinical survival data into statistical models are discussed. We also discuss different penalization approaches for dealing with high dimensionality of high-throughput data to avoid overfitting. Methods are applied to data on ovarian carcinoma, available on the Cancer Genome Atlas, including both clinical and genomic indicator data. Predictive models are fit and used for as making dynamic predictions.

As ever in each of the areas that we have considered, there are further aspect that could be explored and possibly extended. We present some possible avenues for future work in Chapter 6.

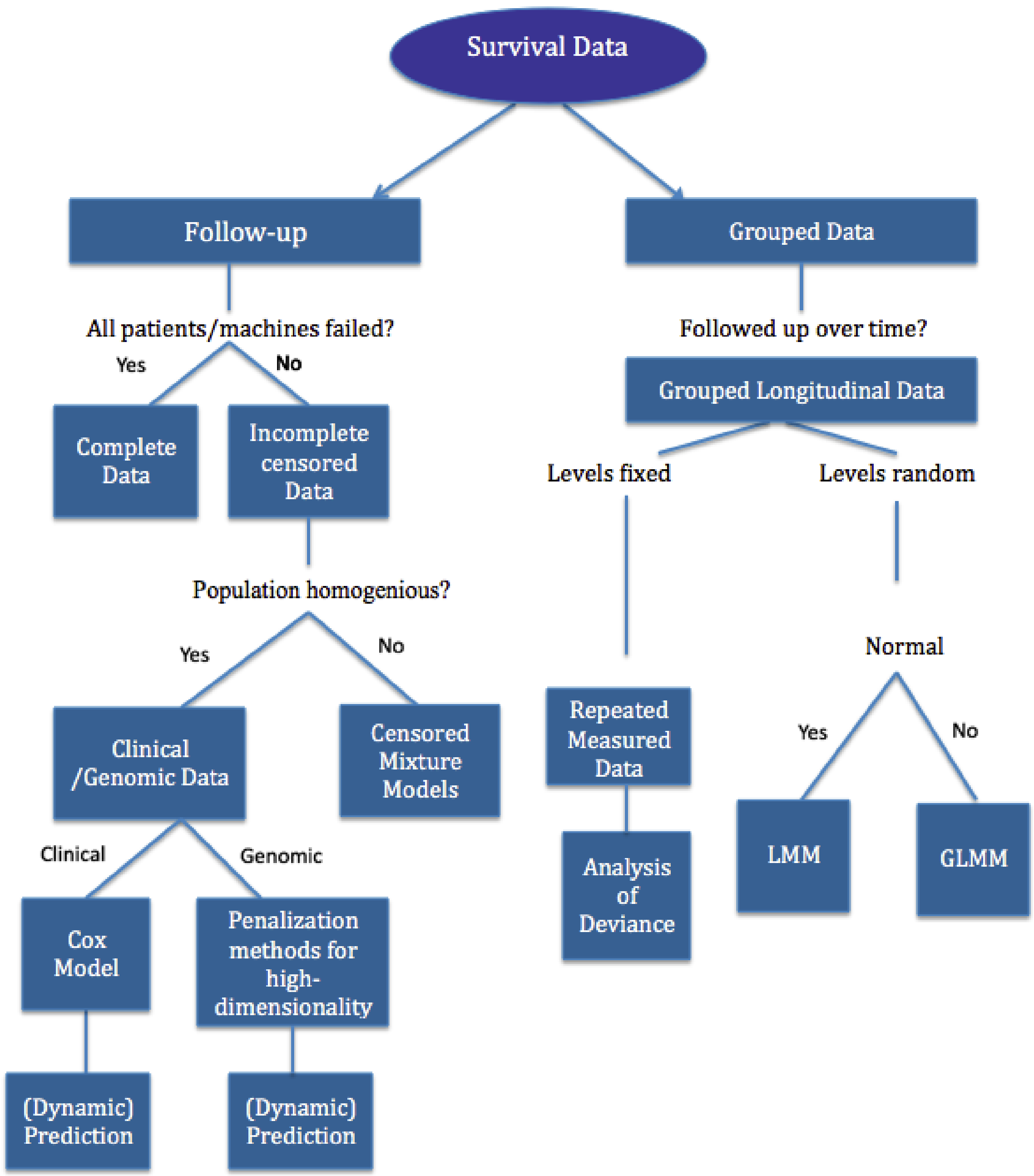


Figure 1.1: Schematic diagram of survival-type data

Chapter 2

Background

The analysis of lifetime or time to event data has been of considerable interest in many branches of statistical applications such as medicine, biological sciences, etc. In other words, the time to an event such as death or recurrence of a tumor, or a positive event such as discharge from hospital areoften of key importance.

Survival data can naturally arise from studies on machines' time to break down, which is also known as reliability, to agricultural experiments (for example how some environmental conditions affect flowering), to medical cohort studies (for example following up cancer patients survival and their reaction to treatments), etc.

2.1 Censored Data

One of the main features of survival data is that during follow up, the event may not usually happen for all the individuals/patients, duo to for example drop-out, intermittent follow-up and finite study duration; in these situations the observations are deemed censored. When an observation is censored it means the information about it is incomplete which can be considered as a form of missing data. In particular, the follow-up time might be such that it prevents full observation and only allows us to observe the survival time of an individual as exceeding a certain amount of time leading to censoring. This kind of censoring also called **"right censoring"**; there are different types of right censoring depending on the experimental condition. Here are two examples:

- Suppose in 20 weeks of follow-up a patient does not experience the event of interest before the end of the study. This patient is said to be right censored and then the survival time is at least as long as the study duration.

- When a person drops out of the study before the end of the study observation time and did not experience the event.

The most common reason for censoring is the need to analyze the time to event data before all the items have failed. It naturally happens in reliability and survival analysis. To acquire moreinformation about different types of censoring one can refer to Balakrishnan and Aggarwala (2000) and Lawless (2011).

Censoring can beplanned in various different ways. Some of the more important censoring schemes are as follows:

1. Type-I censoring
   - Generalized Type-I censoring
   - Progressive Type-I censoring
2. Type-II censoring
   - Progressive Type-II censoring
3. Hybrid censoring
   - Type-I/Type-II hybrid censoring
   - Type-I/Type-II generalized hybrid censoring
   - Unified hybrid censoring
4. Random censoring
5. Interval censoring
   - Bivariate censoring
   - Middle censoring

Some of the above mentioned censoring schemes can beplanned under left censoring as well, however, as the main topic in this thesis is based on the right censoring, they are not discussed here. The most well-known kinds of censoring are Type-I and Type-II censoring schemes which are classified under right censoring. Right censoring schemes of particular importance to this thesis are expanded on in more detail as follows.

2.2 Some famous censoring schemes

Consider a reliability experiment in which a random sample of n identical units are placed on a life-test. Let $X_1, X_2, \ldots, X_n$ denote the lifetimes of these experimental units. As usual, we shall assume that these variables are independent and identically distributed from an absolutely continuous population with cumulative distribution function $F(x)$ and probability density function $f(x)$. Let $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$ denote the corresponding ordered lifetimes, which are the usual order statistics. In life-testing experiments it is, of course, natural for lifetime data to arrive in the form of these order statistics. Considering these assumptions, we introduce the following censoring schemes.

2.2.1 Type-I censoring

In a life-time test, there are situations where for some reasons such as time constraint, the test must be terminated at a particular time t. In this case, all individuals are under observation at time $t = 0$ until one of these two happens:

(a) They fail before time t such that their exact failure times are known.

(b) They work until time t. In this case, due to test termination, the survival time t is recorded for such individuals. This is a censored survival time and the actual survival time is greater than or equal to the test recorded time.

Now suppose r individuals fail prior to time t. We have complete information only on the first $r(\leq n)$ order statistics before time t. Then the data arising from such a time-limited life-test would be of the form $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(r)}$ with all the other $n-r$ lifetimes being more than t but their censored life-times are considered to be equal to t. Here, of course, r is a random variable having a binomial distribution with parameters $(n, F(t))$, $(0 \leq r \leq n)$. This situation is referred to as ***Type-I censoring*** (time censoring). The likelihood function for this type of censoring, following from the probability density function of the first r order statistics, can be written in the following form

$$L(\Theta|\mathbf{x}) = \frac{n!}{(n-r)!} \prod_{i=1}^{r} f(x_{(i)}) \left\{S(t)\right\}^{n-r}, \quad x_{(1)} \leq \ldots x_{(r)} \leq t; 0 < r \leq n,$$

where Θ is the vector of parameters and $\boldsymbol{x} = x_{(1)} \leq \ldots \leq x_{(r)}$ is the vector of order statistics.

2.2.2 Type-II censoring

If the experimenter decides to carry out the life-test until the time of the r-th failure, then the data arising from such a life-test would be of the form $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(r)}$ with the remaining $n-r$ lifetimes being larger than $X_{(r)}$. This situation is referred to as ***Type-II censoring*** (event censoring). The likelihood function for this type of censoring, following from the probability density function of the first r order statistics, is given by

$$L(\Theta|\mathbf{x}) = \frac{n!}{(n-r)!} \prod_{i=1}^{r} f(x_{(i)}) \left\{S(x_{(r)})\right\}^{n-r}, \quad x_{(1)} \leq \ldots \leq x_{(r)}; 0 < r \leq n$$

where Θ is the vector of parameters and $\boldsymbol{x} = x_{(1)} \leq \ldots \leq x_{(r)}$ is the order statistic. If this censoring scheme is applied, one cannot determine the termination point of the experiment; so it can be counted as a weak point of Type-II censoring.

It is clear that there is a fundamental difference between these two censoring schemes that in the case of Type-I censoring, the duration of the life-test is fixed and the number of failures is random, while in the case of Type-II censoring the duration of the life-test is random and the number of failures is fixed.

2.2.3 Progressive Type-II censoring

Progressive Type-II censoring is a natural extension of Type-II censoring which is useful when the loss or removal of units before failures may be unavoidable and they are removed at some prefixed or random inspection times. Herd (1956) were apparently the first to discuss estimation of the population parameters based on progressively censored samples and later, Cohen (1963) discussed the importance of progressive censoring in life-testing reliability experiments.

Under this censoring scheme, from n experimental units that are put on test simultaneously we only observe r complete failures. Given a censoring plan $\mathcal{R} = (R_1, R_2, \ldots, R_r)$:

- At the time of the first failure, i.e. $x_{(1)}$, R_1 units are randomly removed (censored) out the $n-1$ survived units.
- At the time $x_{(2)}$ when the second failure occurs, R_2 units are randomly removed

from the life-test experiment.

- ⋮
- Finally at the time of the last failure, i.e. $x_{(r)}$, all the remaining survived units, $R_r = n - r - \sum_{i=1}^{r-1} R_i$ units are censored.

Notice that under this scheme, censoring occurs progressively in r stages. Figure (2.1) is a schematic picture illustrating this scheme of censoring which is one of the most reproduced figures in the literature on progressive censoring.

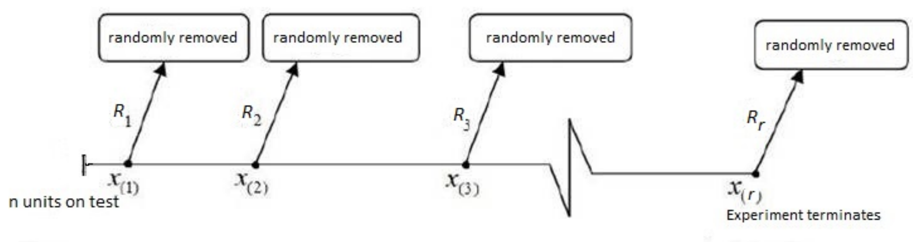


Figure 2.1: Illustration of progressive Type-II censoring

Complete sample situation (with no censoring) and the ordinary Type-II censoring are special cases of this sampling scenario where the first case happens when $r = n$ and $R_1 = \cdots = R_{r-1} = 0$ and the second case happens when the later condition holds.

The likelihood function under this censoring scheme is given by:

$$L(\Theta|\mathbf{x}) = C \prod_{i=1}^{r} f(x_{(i)})\{S(x_{(i)})\}^{R_i}, \quad x_{(1)} \leq \ldots \leq x_{(r)}; 0 < r \leq n,$$

where $C = n(n-R_1-1)(n-R_1-R_2-2)\ldots(n-R_1-\cdots-R_{r-1}-r+1)$. Note that the constant C is not only the normalizing constant, but also represents the number of ways in which the r progressively Type-II right censored samples may occur.

Figure (2.2) shows the histogram of the number of publications on progressive censoring during 1961-2013 which Balakrishnan and Cramer (2014) have obtained by a quick search on zbMATH [1] database.

[1] http://zbmath.org.

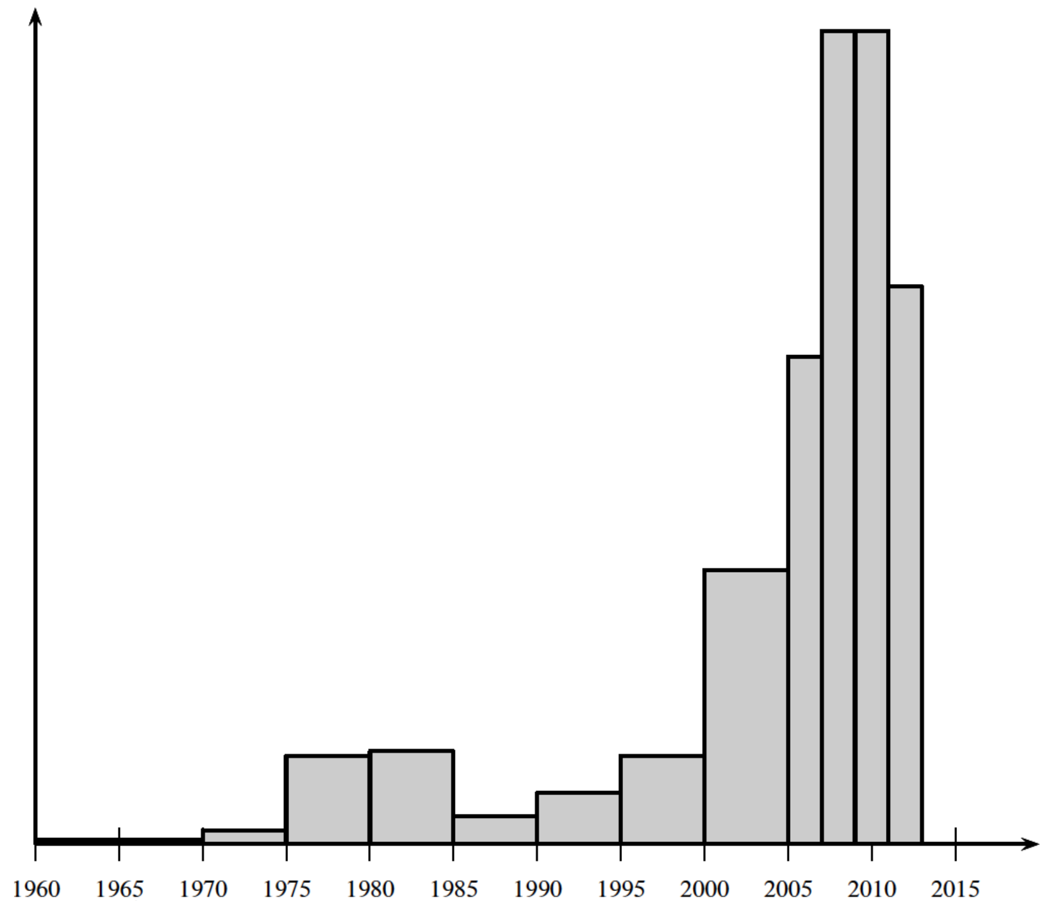


Figure 2.2: More than 50 years of progressive censoring: Histogram of quick search for publications (zbMATH) from 1961-2013.

For the details and most recent developments on progressive Type-II censoring, see Balakrishnan and Aggarwala (2000), Balakrishnan (2007), Balakrishnan and Cramer (2014) and Fallah et al. (2016).

2.3 Mixture models

Mixture models have been at the source of many methodological developments in statistics, as well as providing a flexible environment for statistical modeling while straddling the parametric and the non-parametric approach.

In survival analysis, heterogeneity between individuals is often described by frailty models, where random effects act multiplicatively on the hazard function. This corresponds to a mixture model for survival times.

Discrete mixture models arise when individuals in a population are each one of k distinct types, with a proportion, π_i, of the population being of the i-th type; the π_i satisfy $0 \leq \pi_i \leq 1$ and $\sum \pi_i = 1$. Individuals of type i are assumed to have a lifetime distribution with density function $f_i(t)$ and survivor function $S_i(t)$. An individual randomly selected from this population then hasthe following density

$$f(t) = \pi_1 \ f_1(t) + \cdots + \pi_k \ f_k(t).$$

Hence, according to equation (1.1) the survivor function can be obtained as follows:

$$S(t) = \pi_1 \ S_1(t) + \cdots + \pi_k \ S_k(t). \tag{2.1}$$

Models of this kind are called discrete mixture models, and are useful in situations where the population is heterogeneous but it is not possible to distinguish between individuals of different types, for example susceptible and non-susceptible individuals to failure. Often the $S_i(t)$ in (2.1) are taken to be from the same parametric family, though this is, of course, unnecessary. The properties of a mixture model follow from the properties of the k distributions, or components, involved in the mixture. Estimation can be difficult, and models with k larger than 3 are rarely used.

Suppose $\boldsymbol{\Theta}$ is the vector of unknown parameters. The estimation of $\boldsymbol{\Theta}$ on the basis of the observations $\boldsymbol{y}_j$ is only meaningful if $\boldsymbol{\Theta}$ is ***identifiable***. In general, a parametric family of densities $f(\boldsymbol{y}_j; \boldsymbol{\Theta})$ is identifiable if distinct values of the parameter $\boldsymbol{\Theta}$ determine distinct members of the family of densities

$$\{f(\boldsymbol{y}_j; \boldsymbol{\Theta}) : \boldsymbol{\Theta} \in \Omega\},$$

where $\boldsymbol{\Theta}$ is the specified parameter space; that is,

$$f(\boldsymbol{y}_j; \boldsymbol{\Theta}) = f(\boldsymbol{y}_j; \boldsymbol{\Theta}^*), \tag{2.2}$$

if and only if $\boldsymbol{\Theta} = \boldsymbol{\Theta}^*$.

Identifiability for mixture distributions is defined slightly differently. Suppose that $f(\boldsymbol{y}_j; \boldsymbol{\Theta})$ has two component densities, say, $f_i(\boldsymbol{y}_j; \theta_i)$ and $f_h(\boldsymbol{y}_j; \theta_h)$, that belong to the same parametric family. Then (2.2) will still hold when the component labels i and h are interchanged in $\boldsymbol{\Theta}$. That is, although this class of mixtures may be identifiable, $\boldsymbol{\Theta}$ is not.

Let,

$$f_i(\boldsymbol{y}_j; \boldsymbol{\Theta}) = \sum_{i=1}^{g} \pi_i \; f_i(\boldsymbol{y}_j; \boldsymbol{\theta}_i)$$

and

$$f_i(\boldsymbol{y}_j; \boldsymbol{\Theta}^*) = \sum_{i=1}^{g^*} \pi_i{}^* \; f_i(\boldsymbol{y}_j; \boldsymbol{\theta}_i{}^*)$$

be any two mamebers of a parametric family of mixture densities. This class of finite mixtures is said to be identifiable for $\boldsymbol{\Theta} \in \Omega$ if

$$f(\boldsymbol{y}_j; \boldsymbol{\Theta}) \equiv f(\boldsymbol{y}_j; \boldsymbol{\Theta}^*)$$

if and only if $g = g^*$ and we can permute the component labels so that

$$\pi_i = \pi_i^* \quad \text{and} \quad f_i(\boldsymbol{y}_j; \theta_i) = f_i(\boldsymbol{y}_j; \theta_i^*) \quad (i = 1, \ldots, g). \tag{2.3}$$

Here $\equiv$ implies equality of the densities for almost all $\boldsymbol{y}_j$ relative to the underlying measure on $\mathbb{R}^p$ for $f(\boldsymbol{y}_j; \boldsymbol{\Theta})$.

Identifiability questions for mixtures must be settled before one can meaningfully discuss the problems of parameter estimation of a mixture model consistently on the basis of observations from the mixture. Identifiability makes sure that the correspondence between distributions and mixture parameter vectors is one to one. Lack of identifiability is a serious problem if we intend to classify future observations into one of several classes from the component distributions, for example in Bayesian framework where posterior simulation is used to make inference from the mixture model which is called the label-switching problem. For more details about mixture models, see Teicher (1963), McLachlan and Peel (2004) and Lawless (2011).

Note 2.1. *The operator E_Θ denotes the expectation using the parameter vector Θ wherever it appears in this thesis (to avoid any confusion in taking the expectations).*

2.4 EM algorithm

The Expectation-Maximization algorithm known as the EM algorithm is a general technique for finding maximum likelihood estimates (MLE's) for parametric models when the data are not fully observed, where maximum likelihood estimation is difficult due to the presence of missing data. The EM algorithm was introduced by Dempster et al. (1977) to overcome such difficulties in maximizing likelihoods. In other words, the EM algorithm is an augmentation method for locating the mode or modes of the likelihood function or of the posterior density that augments the observed data with latent data (missing or unobserved data) such that complicated maximizations are replaced by iterative simple maximizations. So the most interesting features of this technique are its simplicity and stability, however, it does not provide standard errors for the parameters.

As the name of the technique suggests, the algorithm consists of two steps in each iteration: E-step and M-step. The E-step involves finding the conditional expectation of the complete-data log-likelihood over the unobserved/latent variables given the current parameter estimates and the observed data. In the M-step the expected complete-data log-likelihood will be maximized over the parameters to find the values of parameters that make the observed data most likely. This process starts from one step with a guess on either the parameter values or the missing values and is done iteratively until convergence where the estimates are not much different from one iteration to the next.

More specifically, let $\boldsymbol{y} = (y_1, \ldots, y_n)$ denote the observed data with density function $f(\boldsymbol{y}|\ \Theta)$. The EM algorithm iterates between a calculation of the expected complete-data likelihood

$$Q(\Theta|\ \hat{\Theta}^{\nu}) = E_{\boldsymbol{u},\hat{\Theta}^{\nu}}\{\log(f(\boldsymbol{y},\boldsymbol{u}|\ \Theta))\}, \tag{2.4}$$

and the maximization of (2.4) over Θ where $\boldsymbol{u} = (u_1, \ldots, u_q)$ is the vector of latent variables, for example can be thought of unknown random effects in a generalized linear model, and $\hat{\Theta}^{\nu}$ denotes the maximum of Θ at the ν-th iteration; It is assumed that $f(\boldsymbol{y}|\ \Theta) = \int f(\boldsymbol{y},\boldsymbol{u}|\ \Theta)\ du$.

If the E-step is difficult to evaluate, we need to implement a Monte Carlo EM algorithm **(MCEM)**. It is a modification of the EM algorithm that calculates the E-step numerically with Monte Carlo simulation. It is also called the **Stochastic EM algorithm**. More precisely, considering the fact that the expectation in (2.4) is over the latent variable $\boldsymbol{u}$, we have

$$E_{\boldsymbol{u},\hat{\Theta}^{\nu}}\{\log(f(\boldsymbol{y},\boldsymbol{u}|\ \Theta))\} = \int \log\left(f(\boldsymbol{y},\boldsymbol{u}|\ \Theta)\right) g\left(\boldsymbol{u} \mid \boldsymbol{y}, \hat{\Theta}^{\nu}\right)\, du$$

where $g\left(\boldsymbol{u} \mid \boldsymbol{y}, \hat{\Theta}\right)$ is the conditional distribution of the latent variables given the observed data and Θ. If we generate a sample $\boldsymbol{u}_1^{\nu}, \ldots, \boldsymbol{u}_m^{\nu}$ from the density $g\left(\boldsymbol{u} \mid \boldsymbol{y}, \hat{\Theta}^{\nu}\right)$, this expectation can be estimated by the Monte Carlo sum

$$Q_m(\Theta \mid \hat{\Theta}^{\nu}) = \frac{1}{m}\sum_{i=1}^{m} \log\left[f(\boldsymbol{y},\boldsymbol{u}_i^{\nu}|\ \Theta)\right]. \tag{2.5}$$

This method makes use of the law of large numbers for proving that the estimator in (2.5) converges to the theoretical expectation in (2.4) when the Monte Carlo sample size m is large enough. A disadvantage of this method is that the accuracy of the results and the speed of convergence are both depend upon the value of m, however, its great advantage is that it can be implemented when E-step of the EM algorithm is hard or impossible. For more details about the implementation of the MCEM one can look at Levine and Casella (2001).

Pros and Cons of the EM Algorithm:

- Provides an stable tool for estimation of parameters in complex incomplete-data problems including finite mixture models.
- Is simple and easy to implement where algorithms such as the Newton-Raphson or gradient decent methods may happen to be too complicated.
- The likelihood value in each iteration is higher than the previous one.
- Convergence is guaranteed only to the local maximum of the objective function.
- Convergence to irrelevant value(s) if the model is non-identifiable.
- Label-switching problem in the estimation of parameters of component densities especially in Bayesian framework when having lack of identifiability.

- It can be painfully slow in some circumstances.
- Covariance matrix is not computed by this algorithm but it can be found in some other ways, such as the Louis method (see the explanation in Section 2.5).

2.4.1 Nested EM algorithm

The EM algorithm is a very popular method for maximum likelihood estimation. Although it does converge, it suffers from taking too long time to do so when missing information is huge or the latent class variables have a complicated structure. The nested EM algorithm is an alternative approach that can be of help to speed up the algorithm. It is basically an EM algorithm in which the M-step is replaced by another EM algorithm called the "Inner EM algorithm" and the expectation and maximization steps that the algorithm starts with form the "Outer EM algorithm". In other words, the nested EM splits the missing data into two parts and handles them appropriately. The advantage of this approach is that the "Inner EM algorithm" is typically fast to converge and leads to an overall faster algorithm and also improves the computational efficiency.

Nested EM algorithm has been used by many authors. Van Dyk (2000) described how nesting two (or more) EM algorithms can take advantage of closed form conditional expectations and lead to algorithms which are faster in convergence with reduced the computational time. In addition the implementation of that is straightforward and enjoys stable convergence (e.g., monotone convergence in likelihood). Also Karlis (2009) used this method when using Exponential Poisson lifetime distribution, a mixture of Exponential distribution with a zero truncated Poisson distribution. For more information on the nested EM algorithm one could refer to Van Dyk (2000), Karlis (2009) and Popescu and Wong (2005).

In this thesis the steps taken in the "Outer EM" are referred to as the First EM Algorithm and those in the "Inner EM" as the Second EM Algorithm where the Second EM Algorithm is based on the mixture representation and the First EM Algorithm handles the censored observations part.

2.5 Louis method for finding the standard errors in presence of latent data

Apart from the fact that the EM algorithm is quite useful with numerous applications in different scenarios, there is a drawback to this method, it does not provide the variance-covariance matrix of the parameter estimates while some other standard optimisation tools do, for example, the function "`optim`" in `R` and the package "`bbmle`" can be mentioned which gives some tools for maximum likelihood estimation such as the function "`mle2`" with an option for returning the Hessian matrix (see Bolker and Team, 2014).

Among all existing methods, such as Louis (1982), Baker (1992) and Oakes (1999), for finding the standard errors when using the EM algorithm, Louis (1982) method seems to be the most popular one. The main idea in this technique is that the observed information matrix is computed on the last iteration of the EM algorithm using only the gradient vector, also known as score statistic, and the second derivative matrix of the complete-data log-likelihood which are generally easier to work out than those of incomplete-data log-likelihood.

According to the Louis' method, the observed information matrix is defined as

$$I_y(\Theta) = E_\Theta\Big\{B(x;\Theta|x)\Big\} - E_\Theta\left\{S\left(x;\Theta|x\right)S^T\left(x;\Theta|x\right)\right\} + S^*(y;\Theta)S^{*T}(y;\Theta), \tag{2.6}$$

where $S(x;\Theta)$ and $S^*(y;\Theta)$ are the gradient vectors of the complete and incomplete-data log-likelihood, respectively; $E_\Theta\Big\{B(x;\Theta|x)\Big\}$ is the expectation of the negative of the 2nd derivative of the complete-data log-likelihood, being the so-called Fisher Information (FI); and $S^*(y;\Theta) = E_\Theta\{S(x;\Theta|x)\}$. Notice that $S^*(y;\hat{\Theta}) = 0$. It uses the following relation

$$\text{Observed information} = \text{Complete information} - \text{Missing information}, \tag{2.7}$$

based on the Missing Information Principle.

The second way of applying the Louis' method is given in (McLachlan and Krishnan, 2007, page 132) as

$$I_y(\hat{\Theta}) = \mathcal{I}_c(\hat{\Theta}; y) - \Big[\text{cov}_{\Theta}\left\{S(x;\Theta)|y\right\}\Big]_{\Theta=\hat{\Theta}}, \tag{2.8}$$

where $\mathcal{I}_c(\hat{\Theta}; y)$ denotes the expected (conditional) complete-data information matrix. The latter is defined as

$$\mathcal{I}_c(\hat{\Theta}; y) = E_{\Theta}\left\{-\partial^2 \log L_c(\Theta)/\partial\Theta\ \partial\Theta^T|y\right\},$$

where the subscript c corresponds to complete-data.

Remark. *Both methods in (2.6) and (2.8) are essentially equivalent and can be used as appropriate.*

As an alternative way of finding the expected FI under progressive Type-II censoring, we summarize the method introduced by Ng et al. (2002) as follows, which basically springs from the missing information principle, (see Tanner, 1991; Louis, 1982).

Let $\boldsymbol{X}^{\mathcal{R}} = (X_1, X_2, \ldots, X_r)$ be the observed progressively Type-II censored order statistic sample from $\boldsymbol{T} = (T_1, T_2, \ldots, T_n)$ and consider that as the observed information. Also let $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_r)$ denote the censored random variables where $\boldsymbol{Z_j} = (Z_{j1}, \ldots, Z_{jR_j})$ is the random variable corresponding to the censored units at the j-th censoring stage, i.e., the j-th failure time, which deems to be the missing data. Combining $\boldsymbol{X}^{\mathcal{R}}$ and Z gives the complete data set $\boldsymbol{T}$. Now consider the following theorem by Ng et al. (2002) which is needed for finding the FI later on.

Theorem 2.1. *Given* $\boldsymbol{x_j}^{\mathcal{R}} = (x_1, \ldots, x_j)$, *the conditional density of* $Z_{jk}, k = 1, \ldots, R_j$; $j = 1, \ldots, r$, *is given by*

$$f^{Z_{jk}|\boldsymbol{X_j}^{\mathcal{R}}}(z_{jk}|x_j^{\mathcal{R}}) = f^{Z_{jk}|X_j}(z_{jk}|x_j) = \frac{f(z_{jk})}{1 - F(x_j)}, \quad z_{jk} > x_j \tag{2.9}$$

and Z_{jk} *and* Z_{jl}, $k \neq l$, *are conditionally independent given* x_j.

Proof. See (Ng et al., 2002, page 373) or (Balakrishnan and Cramer, 2014, page 207).

Now we can define the information for observed, complete and missing data which are denoted respectively as $\mathcal{I}\left(\boldsymbol{X}^{\mathcal{R}};\theta\right)$, $\mathcal{I}\left(\left(\boldsymbol{X}^{\mathcal{R}},\boldsymbol{Z}\right);\theta\right)$ and $\mathcal{I}\left(\boldsymbol{Z}|\boldsymbol{X}^{\mathcal{R}};\theta\right)$ as follows.

First,

$$\mathcal{I}\left(\left(\boldsymbol{X}^{\mathcal{R}},\boldsymbol{Z}\right);\theta\right) = -E\left[\frac{\partial^2}{\partial\theta^2}\log f^{\boldsymbol{X}^{\mathcal{R}},\boldsymbol{Z}}(\boldsymbol{X}^{\mathcal{R}},\boldsymbol{Z})\right] = \mathcal{I}(\boldsymbol{T};\theta) = n\,\mathcal{I}(X_1;\theta).$$

Now defining

$$\mathcal{I}^{(j)}(\boldsymbol{Z_j}|X_j;\theta) = -E\left[\frac{\partial^2}{\partial\theta^2}\log f_\theta^{\boldsymbol{Z}_j|X_j}\left(\boldsymbol{Z_j}|X_j\right)\right],$$

for the random variable $\boldsymbol{Z}$ progressively censored at the time of the j-th failure, we arrive at the expected FI of $\boldsymbol{Z}$ given $\boldsymbol{X}^{\mathcal{R}}$ as below:

$$\mathcal{I}\left(\boldsymbol{Z}|\boldsymbol{X}^{\mathcal{R}};\theta\right) = \sum_{j=1}^{r} R_j\,\mathcal{I}^{(j)}\left(\boldsymbol{Z_j}|X_j;\theta\right). \tag{2.10}$$

The above two relations are related using the missing information principle (2.7). Hence, the expected FI can be found from

$$\mathcal{I}\left(\boldsymbol{X}^{\mathcal{R}};\theta\right) = \mathcal{I}\left(\boldsymbol{T};\theta\right) - \sum_{j=1}^{r} R_j\,\mathcal{I}^{(j)}\left(\boldsymbol{Z_j}|X_j;\theta\right). \tag{2.11}$$

Note 2.2 (Regularity Conditions)**.** *All the methods mentioned above hold under the Regularity Conditions. Suppose θ is the location or scale parameter, the function f_θ satisfies the regularity conditions if:*

(i) $\forall x \in \mathbb{R}, f_\theta(x) > 0,$

(ii) $\lim_{x\to\infty} x^2 f_\theta'(x) = 0$ *where f_θ' is the derivative of f_θ,*

(iii) The second derivative f_θ'' is continuous,

(iv) The expectation $\int -\frac{\partial^2 \ln(f_\theta(x))}{\partial\theta^2} f_\theta(x)\,dx$ is finite, where f_θ denotes the density from the location or scale family.

2.6 Moments and Delta Method

Moments

A well-known feature that is usually used for identifying and describing a distribution is the distribution's moment generating function.Note that the moment generating function for a random variable Y is defined as $M_Y(t) = E\left[e^{ty}\right]$.

For the Normal distribution which is used in this chapter, the moment generating functions for the random variables $Y \sim N(\mu, \sigma)$ and $Y|Y > a$ are given respectively by

(i) $M_Y(t) = \exp\{\mu t + \frac{\sigma^2 t^2}{2}\}$

(ii) $M_{Y|Y>a}(t) = \exp\{\mu t + \frac{\sigma^2 t^2}{2}\} \frac{1-\Phi(\frac{a-\mu-\sigma^2 t}{\sigma})}{1-\Phi(\frac{a-\mu}{\sigma})}$,

where a is an arbitrary constant.

Delta Method

In certain situations statisticians encounter a problem of developing an expression for the variance of the estimator of interest for purposes such as confidence interval estimation. The delta method is a common way for finding an approximate expression when the estimator is not simply a sum of observations. The idea of this method is using the so-called Taylor Series Expansion to find a linear function that approximates the more complicated functions. So the delta method can be used only when the function of interest can be approximated by a Taylor series, i.e. a smooth function with no corners. The delta method uses the first two terms of the series about the mean of the variable to approximate the value of the function as follow:

$$f(X) \cong f(\mu) + (X - \mu)f'(\mu)$$

where $f'(\mu) = \frac{\partial f(X)}{\partial X}|_{X=\mu}$. From the above we can derive the variance of f approximately as:

$$\begin{aligned} Var(f(X)) &\cong Var(X - \mu)\left[f'(\mu)\right]^2 \\ &\cong \sigma^2 \times \left[f'(\mu)\right]^2 \end{aligned} \tag{2.12}$$

The delta method estimator of the variance of the function is derived when μ and σ in (2.12) are replaced by their estimators as follows:

$$\widehat{Var(f(X))} = \hat{\sigma}^2 \times \left[f'(\hat{\mu})\right]^2 \tag{2.13}$$

An example of applying this method, can be found in equation (3.28).

2.7 Longitudinal Data

Longitudinal data which are also known as *panel data*, are commonly used in medical research designs where patients baseline health status are measured, an intervention is undertaken, and follow-up measurements are collected. This yields a repeated measured response variable over time for multiple subjects. To be more specific, a longitudinal study refers to an investigation where subjects outcomes and exposures are collected at multiple times in a follow-up where observations within-subjects are possibly correlated and there are possible heterogeneity between-subjects. Notice that in a longitudinal study, the response variables can be either continuous or discrete.

Example 2.1. HIV patients may be followed over time and clinical factors such as CD4 counts are measured monthly to characterize their immune status.

In many longitudinal studies both repeated measurements information and the time to a clinical event such as death or recurrence are collected.Hence, longitudinal data can be deemed as time to event data when multiple events happen to an individual over time. Analysis of such type of data fits in the context of *survival analysis*. Possible statistical analysis approaches, parametric and non-parametric, are discussed substantially in Chapter 4 for the analysis of longitudinal data which is grouped/clustered over time intervals with application to a real data.

The aim of a statistical analysis of longitudinal data is often modeling the expected value of the response variable as a function of explanatory variables which could be either linear or nonlinear. Typical statistical methods includes generalized linear models and generalized linear mixed models. They come with standard link functions defining a relation between the mean response and the linear form of explanatory variables. These models are introduced in the next two sections.

For information on the benefits and challenges of longitudinal data, one could see (Van Belle et al., 2004, Chapter 18).

2.8 Generalized Linear Models

The widely used Generalized Linear Model was first introduced by Nelder and Baker (1972) which generalizes the linear models, as the name reflects. The essence of this generalization is that first, the data are not always normal as a linear model requires and second, the mean response is not necessarily a linear function of the parameters but a function of the mean may be.

In building a generalized linear model, the response is assumed to be a collection of independent measurements from a distribution of the exponential family or similar to that, for example Normal, Poisson or binomial. Then a proper link function is considered as to link the mean (expected value) of the response and the linear form of explanatory variable which are also called predictors. Applying a link function makes the expected value of the response linear in terms of the predictors and also makes its expected variance homogeneous. The structure is as follows:

$$
\begin{aligned}
E(y_i) &= \mu_i \\
g(\mu_i) &= \boldsymbol{x}_i'\boldsymbol{\beta}
\end{aligned}
$$

where $g(\cdot)$ is the link function, $\boldsymbol{x}_i'$ is the i-th row of the model matrix, and $\boldsymbol{\beta}$ is the parameter vector.

The standard link functions include, *probit*, *logit*, *log*, *identity*, *reciprocal*, etc. For more information, see Faraway (2006) or Dobson and Barnett (2008).

2.9 Generalized Linear Mixed Models

When data are classified in terms of factors with a number of levels, we are often interested in exploring to what extent each level of the factors influence the variable of interest. This is referred to as the *effect* of a level of a factor which is usually represented in terms of a parameter. If the parameters are fixed constants, they are called *fixed effects* and if for example there are infinite levels of a factor from which only a random sample occur in the data, they are called *random effects* – in other words, when levels are sampled from a larger population, or the interest lies in the variation among the whole levels of the factor rather than the specific levels.

When some of the parameters in a generalized linear model are treated as random and some as fixed, the model is called a *generalized linear mixed model*. It basically combines the properties of two statistical frameworks, (i) linear mixed models: as it incorporates random effects into the model and (ii) generalized linear models: as it can handle non-normal data by using link functions and exponential family distributions. These features make the generalized linear mixed models the best tool for analyzing non-normal data with random effects, however, they are surprisingly challenging to use even for statisticians, as mentioned by Bolker et al. (2009).

Estimation of parameters is an integral part of statistical models. In generalized linear mixed model, the parameters to be estimated are the fixed-effect parameters including effects of covariates, differences among treatments and interactions and random-effect parameters including the standard deviations of the random effects. For more detailed information on generalized linear mixed models see Lee et al. (2006), McCulloch and Neuhaus (2001) or Molenberghs and Verbeke (2004).

2.10 Predictive Modeling

In medical applications of survival analysis, sometimes there is a need to extrapolate the survival function, defined in (1.1), to make a prediction for patients and answer the question "how long will a patient live". This question is not only asked by patients at the beginning of the treatment, but also at every visit.Therefore, a number of hospitals save the clinical and genomic data from patients to be used for the purpose of building predictive models and hence helping doctors use the best (individual) treatment based on the patient's diagnosis and prognosis.

A traditional method for prediction beyond the time window of available data, is left truncation. It consists of taking into consideration only patients who were alive as of some point in the past (truncation date).

$$S(s|t) = P(T > s|T \geq t) = \frac{S(s)}{S(t^-)}, \quad s \geq t \tag{2.14}$$

where S denote the survival function and t^{-} indicates the time just before the truncation time t.

This method does not seem to be very accurate as the clinical factors of patients are changing dynamically over time. In Chapter 5 of this thesis, we look at methods for dynamic prediction based on the survival function and apply it to a real data. Notice that predictive ability and accuracy of a model needs to be assessed and evaluated which requires certain methods.

2.11 Microarray Gene Expression Data

Microarray is a revolutionary technology that allows measuring of genetic expressions for tens of thousands of genes in a cell or tumor sample simultaneously.

One of the most interesting envisioned application of microarray technology has been its use in the clinic. Microarray technology provides fresh understanding of the disease processes at the molecular level, which would allow us to improve our understanding of diseases and help explore new subtypes, from which the idea of *personalized medicine* has sprung where diagnosis and prognosis, and also treatments, would depend on the patients' genetic information.

Methods for detecting the associations between gene expression profiles and phenotypic data has been explored already in the literature, including predictions for new samples based on such relationships, for example see Whitfield et al. (2003). However, Park (2005) noted that categorizing the survival data to a short-term against long-term survival indicator does not provide an efficient use of data. Extensive studies investigated the correlation between distinct expression profiles and many disease types and this has led to much insight into the biological mechanism underlyingthose diseases, nonetheless, a more adequate way to show their usefulness for patient care is by exploring the links between gene expression profiles and patient survival. However, integrating expression profiles with survival data is challenging due to presence of censoring which is the main difficulty.

Integrating gene expression data with other biomedical resources will provide new mechanistic or biological hypotheses. However, any correlation between survival data analysis and analysis of gene expression naturally inherits all the associated problems with high-dimensional datasets as well as the problems caused by censoring. Therefore, innovative statistical techniques and computing algorithms are essential for the successful analysis of microarray data.

In Chapter 5 of this thesis, we deal with statistical techniques for genetic gene expression data and clinical data and their promising applications for analyzing data from microarray experiments.

Further reading on the wide range of survival analysis of gene expression profiles and the integrated analysis of survival datasets and gene expressions can also be found in Cancer Genome Atlas Research Network and others (2011), Shoemaker and Lin (2004), van Houwelingen et al. (2006), van Houwelingen and Putter (2011), Gui and Li (2005), Jung et al. (2005) and Van De Vijver et al. (2002).

2.12 Conclusions

In this chapter we have introduced the background to survival data and other basic methodological concepts that are necessary for the development in the other chapters of this thesis. The aim has been to provide sufficient detail to clarify concepts for those who may be unfamiliar with them, but without needless repetition of details from the literature.

In the following chapters, we discuss methods and difficulties of the modeling and analysis of different types of survival data that arise in practice from different application areas.

Chapter 3

Modeling Homogeneous and Heterogeneous Populations under Type-II and Progressive Type-II Censoring

3.1 Introduction

As mentioned in Chapter 2, here we deal with the analysis of life-time data with different types of censoring. Many different probability models have been proposed for survival times with particular approaches being popular in different fields of application. For example, Lawless (2011) surveys many different probability models to represent lifetime distributions and describes inference based on those models for lifetime data in the presence of censoring. Historically, the famous Cox (1972) paper introduced a semi-parametric proportional hazards model to analyze lifetime data based on the assumption of independence between the survival times. Earlier Gupta (1952) found the maximum likelihood estimates and best linear unbiased estimates (BLUEs) for the normal model under Type-I and Type-II censoring. More recent work includes Rasouli and Balakrishnan (2010), which develops the exact likelihood inference for two exponential populations under joint progressive Type-II censoring, Park and Lee (2012), derives the maximum likelihood estimates for a homogeneous right censored sample from several distributions, such as the normal and Laplace, using an EM algorithm and a stochastic EM approach and Fallah et al. (2016) which obtains the maximum likelihood, Bayes estimates and also robust posterior regret gamma minimax estimators over three different prior classes, based on a progressively Type-II censored sample from a Pareto population.

Many of the above modeling approaches include allowing for individual subject, or unit, variability by the inclusion of explanatory variables in the model. However, there are situations where we may have heterogeneous populations due to some unmeasured, or latent, characteristic. For example, one can think of patients as being susceptible and non-susceptible to the condition under study with non-susceptible patients having reduced hazard and, on average, longer survival times. One possible approach is to use a finite mixture structure to capture this heterogeneity, which has the additional advantage that it allows the clustering of individuals to the different groups together with parameter estimation for the mixture components. Such models are commonly fitted using the EM algorithm, an iterative algorithm where the component labels in the mixture model are treated as latent/missing variables. For complete data, with no censoring, this is a standard finite mixture model for a positive random variable, the survival times. Everitt and Hand (1981), Tanner (1991) and McLachlan and Peel (2004) are references on mixture models, the EM algorithm and their relationship.

This becomes more complicated in the presence of censoring and requires care in model fitting and interpretation. Kuo and Peng (2000) employed a mixture model to analyze survival data in the presence of right and interval censoring including covariates, using the EM algorithm, or Monte Carlo EM (MCEM), for maximum likelihood estimation and Markov Chain Monte Carlo (MCMC) methods for Bayesian inference. Their proposed method is illustrated with a real data example on male beetles' lifetimes after exposure to toxicity. They consider six different models and show that a normal-exponential mixture model fits the best after doing 2-fold and 4-fold cross validations. Some other recent references on mixture modeling of censored survival data are Li and Lin (2009), Santos and Achcar (2011) and Ateya (2014).

Progressive Type-II censoring is a natural extension of Type-II censoring where a number of live units are censored at each failure time; so the censoring is progressive over the duration of the study rather than all at the end, as in standard Type-II censoring. In Type-II censoring we determine the duration of the study based on waiting to observe a fixed, pre-specified, number of failures, r, hence the actual duration of a Type-II censored study is random, unlike under Type-I censoring where the study time is fixed with censoring occurring for those units that have not failed by this time. Notice that the number of censored units at each progressive censoring stage can be a random variable or a pre-determined number. However, here we determine the number of removals for the first $r-1$ censoring stages at random by drawing from a Binomial distribution and at the last failure time the number of removals is simply the number of remaining surviving units.

We first consider a single two component mixed normal population and implement a (progressive) Type-II censoring scheme. In order to study the effect of censoring schemes on estimation of parameters and their accuracy and also compare the censoring plans, we do this by considering a simulation study with six comparatively exploratory cases for each censoring scheme. In this chapter:

- First, we assume having complete-data with known labels and no censoring and estimate the parameters and their standard errors. This gives a baseline for complete information.
- Then, we consider a normal population with known labels and find the results after applying a progressive Type-II and Type-II censoring schemes considering the impact of certain numbers of failures defining the test termination point.
- We then consider a mixed normal population (unknown labels), initially with no censoring.
- Finally, we consider a mixed normal population and study the effects of Type-II and progressive Type-II censoring.

Parameter estimates in all these cases are found with two methodologies, (i) direct optimisation using the `R` function `optim` and (ii) the EM algorithm that we have developed. Results of simulation studies which are carried out for all these settings are summarized along with conclusions.

As a general set-up for all the undergoing cases in this chapter, we have the following assumptions:

1. Failure times are independently distributed from a population of size N that comprises of two groups.

2. $\boldsymbol{X} = (X_1, \ldots, X_m)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$, $N = m + n$, are i.i.d. random variables representing the lifetimes of the first and the second group under study with probability density functions f_1 and f_2, respectively. Here, for a specific study, we take these distributions to be $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, respectively, as the normal distribution is one of the general purpose distributions that is also used as a lifetime distribution. Notice that the Exponential life-time distribution, with a long tail on the right, has been studied before for censored mixture observations, but Normal distribution has not been used before in this regard. By applying the same approach to log-transformed failure times we would be using a log-normal distribution and the general approach can also be extended to other component densities.

3. $z_1, \ldots, z_N$ are lifetime observations when whether the individual belongs to either $\boldsymbol{X}$ or $\boldsymbol{Y}$ is not known, i.e., when observations are not labeled. Also let $Z_1 \leq \ldots \leq Z_N$, denote the order statistics of the combined set of all observations from the two groups, i.e., $\boldsymbol{X} = (X_1, \ldots, X_m)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ which are referred to as mixture components. Notice that the censored version of data can be formed based on the order statistics, which is discussed in Section 3.4.1.

4. When censoring occurs, the censored true lifetimes of the X and Y type, are denoted by x^* and y^* respectively.

5. In regard to the censoring schemes implemented in simulation studies, n_1 and n_2 represent the number of censored observations from the first and second component respectively. Also, r_1 and r_2 indicate the number of failures from the respective first and second mixture components.

6. When there is need to use different notation for clarity, this is explained in the relevant section.

To this end, in the next section we propose a proposition, along with its proof, which is necessary for computations in subsequent sections. Following that, we explore the aforementioned scenarios in both theory and through simulations.

3.2 Moments of a right-censored normal distribution

Proposition: For a random variable $Y \sim N(\mu, \sigma^2)$, if it is right censored at a, the resulting random variable $Y|Y > a$ has moment generating function $M_{Y|Y>a}(t) = \exp(t\mu + \frac{1}{2}\sigma^2 t^2)\frac{\overline{\Phi}\left(\frac{1}{\sigma}\left[a-(\sigma^2 t+\mu)\right]\right)}{\overline{\Phi}\left(\frac{a-\mu}{\sigma}\right)}$. Hence, the raw conditional moments $E\left(Y^k|Y > a\right)$, $k = 1, 2, 3, 4$, are then given by the following formulae:

$$\text{(i)}\quad E\left(Y\ |Y > a\right) = \mu + \sigma\ h\left(\frac{a-\mu}{\sigma}\right)$$

$$\text{(ii)}\quad E\left(Y^2|Y > a\right) = \mu^2 + \sigma^2 + \sigma(\mu + a)\ h\left(\frac{a-\mu}{\sigma}\right)$$

$$\text{(iii)}\quad E\left(Y^3|Y > a\right) = \mu^3 + 3\mu\sigma^2 + \sigma^3 + \left\{2\sigma^3 + \sigma(a^2 + \mu^2 + \mu a)\right\} h\left(\frac{a-\mu}{\sigma}\right)$$

$$\text{(iv)}\quad E\left(Y^4|Y > a\right) = \mu^4 + 3\sigma^4 + 6\mu^2\sigma^2 + \{7\sigma^3\mu + \mu^3\sigma + \sigma^3 a + \mu^2\sigma a + \mu\sigma a^2 + \sigma a^3\}\ h\left(\frac{a-\mu}{\sigma}\right)$$

where a is an arbitrary constant, and $\overline{\Phi}(\cdot) = 1 - \Phi(\cdot)$ and $h(\cdot) = \frac{\Phi(\cdot)}{1-\Phi(\cdot)}$ are survival and hazard functions of the standard normal distribution respectively.

Proof of Proposition.

(i).

$$\frac{\partial M_{Y|Y>a}(t)}{\partial t} = (\mu + \sigma^2 t)\exp\{\mu t + \frac{\sigma^2 t^2}{2}\}\frac{S(\frac{a-\mu-\sigma^2 t^2}{\sigma})}{S(\frac{a-\mu}{\sigma})} + \left\{\frac{\sigma\Phi(\frac{a-\mu-\sigma^2 t^2}{\sigma})}{1-\Phi(\frac{a-\mu}{\sigma})}\right\}$$

Therefore, evaluating this at $t = 0$, $E(Y|Y > a) = \mu + \sigma\ h(\frac{a-\mu}{\sigma})$.

(ii).

$$\begin{aligned}\frac{\partial^2 M_{Y|Y>a}(t)}{\partial t^2} &= \left[\sigma^2 + (\mu + \sigma^2 t)^2\right]\exp\{\mu t + \frac{\sigma^2 t^2}{2}\}\frac{S(\frac{a-\mu-\sigma^2 t^2}{\sigma})}{S(\frac{a-\mu}{\sigma})} \\ &\quad + \left[2(\mu + \sigma^2 t) + \Phi(\frac{a-\mu-\sigma^2 t}{\sigma})\right]\exp\{\mu t + \frac{\sigma^2 t^2}{2}\}\left\{\frac{\sigma\Phi(\frac{a-\mu-\sigma^2 t}{\sigma})}{1-\Phi(\frac{a-\mu}{\sigma})}\right\}\end{aligned}$$

So one can find, $E(Y^2|Y > a) = \mu^2 + \sigma^2 + \sigma(\mu + a)\ h(\frac{a-\mu}{\sigma})$.

(iii).

$$\begin{aligned}
\frac{\partial^3 M_{Y|Y>a}(t)}{\partial t^3} &= \Big\{\sigma^2(\mu+\sigma^2 t)+2\sigma^2(\mu+\sigma^2 t)+(\mu+\sigma^2 t)^3\Big\} \\
&\times \exp\{\mu t+\frac{\sigma^2 t^2}{2}\}\frac{S(\frac{a-\mu-\sigma^2 t^2}{\sigma})}{S(\frac{a-\mu}{\sigma})}+\Big\{\sigma^4+3\sigma(\mu+\sigma^2 t)^2 \\
&+2\sigma^3+\sigma(\mu+\sigma^2 t)(a-\mu-\sigma^2 t)-\sigma^3+\sigma\big[(a-\mu-\sigma^2 t) \\
&\times(\mu+\sigma^2 t)+(a-\mu-\sigma^2 t)^2\big]\Big\}\exp\{\mu t+\frac{\sigma^2 t^2}{2}\} \\
&\times\left\{\frac{\Phi(\frac{a-\mu-\sigma^2 t}{\sigma})}{1-\Phi(\frac{a-\mu}{\sigma})}\right\}
\end{aligned}$$

After simplifying the above formula and substituting $t = 0$, we get $E(Y^3|Y > a) = 3\mu\sigma^2 + \sigma^3 + \{2\sigma^3 + \sigma(a^2 + \mu^2 + \mu a)\}\ h(\frac{a-\mu}{\sigma})$.

(iv).

$$\begin{aligned}
\frac{\partial^4 M_{Y|Y>a}(t)}{\partial t^4} &= \Big[3\sigma^4+6\sigma^2(\mu+\sigma^2 t)^2+(\mu+\sigma^2 t)^4\Big]\exp\{\mu t+\frac{\sigma^2 t^2}{2}\} \\
&\times\frac{S(\frac{a-\mu-\sigma^2 t^2}{\sigma})}{S(\frac{a-\mu}{\sigma})}+\Big[8\sigma^3(\mu+\sigma^2 t)+\sigma^3(a-\mu-\sigma^2 t) \\
&+4\sigma(\mu+\sigma^2 t)^3+6\sigma^3(\mu+\sigma^2 t)^2(a-\mu-\sigma^2 t) \\
&+6\sigma(\mu+\sigma^2 t)(a-\mu-\sigma^2 t)^2+\sigma(a-\mu-\sigma^2 t)^3\Big] \\
&\times\exp\{\mu t+\frac{\sigma^2 t^2}{2}\}\left\{\frac{\Phi(\frac{a-\mu-\sigma^2 t}{\sigma})}{1-\Phi(\frac{a-\mu}{\sigma})}\right\}
\end{aligned}$$

Therefore, $E(Y^4|Y > a) = \mu^4 + 3\sigma^4 + 6\mu^2\sigma^2 + \{7\sigma^3\mu + \mu^3\sigma + \sigma^3 a + \mu^2\sigma a + \mu\sigma a^2 + \sigma a^3\}\ h(\frac{a-\mu}{\sigma})$.

3.3 Complete Normal data

We generated a complete data set with the assumption of known observation labels from both $\boldsymbol{X}$ and $\boldsymbol{Y}$ groups and consider the model with the likelihood $L = \prod_{i=1}^{m} f_1(x_i)\times \prod_{j=1}^{n} f_2(y_j)$ to find the maximum likelihood estimate of the parameters μ_1, σ_1, μ_2 and σ_2 and their associated standard errors as well as the proportion of individuals from each group and the test time duration. See the results in section

3.7.4.

3.4 Labeled (progressive) Type-II censored data

3.4.1 Model

Suppose that we have N items from a finite mixture of $N(\mu_j, \sigma_j)$, $j = 1, 2$, components with known labels. All items/individuals are put on a life testing experiment simultaneously and the failure times are observed. The i-th failure time is shown by x_i, if i-th failure is of X type and by y_i, if of Y type; while x_i^* and y_i^* denote the censored life times of the X and Y type, respectively. In other words, x_i^* and y_i^* are the unobserved true failure times that can be estimated.

If the plan of censoring is Type-II censoring, then x_i^* and y_i^* denote the censored life times of the X and Y type, immediately after observing the r-th failure, i.e. the unobserved true failure times which are definitely greater than the time of the r-th failure, Z_r. Notice that under Type-II censoring, $Z_1 \leq \ldots \leq Z_r = Z_{r+1} = Z_{r+2} = \ldots = Z_N$ and the censoring time, i.e. the r-th failure time, can be considered a random variable, from a suitable distribution, while the number of observed failures, r, is fixed. We hereafter refer to $\boldsymbol{Z}$ as the censored version of the data.

Under progressive Type-II censoring, we again assume that the experiment is continued until observing a prefixed number of failures, r, however, immediately after each failure some live units are removed from the experiment according to a censoring plan $\mathcal{R} = (R_1, R_2, \ldots, R_r)$. Based on this censoring scheme, R_i units are withdrawn after the i-th failure such that

$$R_i \sim Bin\left(N - i - \sum_{j=1}^{i-1} R_j, p\right) \quad \text{and} \quad R_r = N - r - \sum_{j=1}^{r-1} R_j.$$

It is worth mentioning that p is the parameter controlling the desired amount of censoring at each progressive censoring stage. To be more specific, p is the probability of being censored for each individual at each observed failure. Hence, for large values of r, the value of p cannot be chosen too large as otherwise most of remaining individuals are likely to be censored and this may make it impossible to obtain the desired number of observed failures, r. Notice that this implies that the condition

$N - i \geq \sum_{j=1}^{i-1} R_j$, $\forall i$, $i = 1, \ldots, r$, must always hold to ensure that at each failure time sufficient individuals remain to enable the full number of r failures to be observed. Other strategies that could be considered include (i) a fixed $\mathcal{R}$ plan rather than random, (ii) depending p on i, censoring stage, for example through considering $\{p_i\}$ a decreasing sequence.

Notice that with this form of data, we could form a likelihood to account for censored observations and use direct optimization (such as `optim` function in `R`) to fit the model. However, instead, we are going to take advantage of the EM algorithm which can be used to estimate the censored observations as in Aitkin et al. (2009).

In order to develop the complete-data likelihood, we make use of the concept of missing data suggested in McLachlan and Krishnan (2007) and represent the missing data by defining the variable $\boldsymbol{C} = (C_1, \ldots, C_N)$ as the known component indicator and the variable $\boldsymbol{W} = (W_1, \ldots, W_N)$ as the censoring indicator. Notice that W is also observed. C_i and W_i, for the i-th observation, are defined as follows:

$$C_i = \begin{cases} 1 & i\text{-th observation} \in \text{group } 1 \\ \\ 0 & i\text{-th observation} \in \text{group } 2 \end{cases} \qquad i = 1, 2, \ldots, N \tag{3.1}$$

and

$$W_i = \begin{cases} 1 & i\text{-th observation is not censored.} \\ \\ 0 & i\text{-th observation is censored.} \end{cases} \qquad i = 1, 2, \ldots, N. \tag{3.2}$$

Now the complete-data likelihood can be written as

$$L \propto \prod_{i=1}^{N} \left\{ \big(f_1(x_i)\big)^{W_i} \ \ \big(f_1(x_i^*)\big)^{1-W_i} \right\}^{C_i} \left\{ \big(f_2(y_i)\big)^{W_i} \ \ \big(f_2(y_i^*)\big)^{1-W_i} \right\}^{1-C_i} \tag{3.3}$$

Generally it is easier to work with the log-likelihood. From (3.3), one can write the log-likelihood for the i-th observation as $l_i = C_i \ l_1^* + (1 - C_i) \ l_2^*$. Hence, the complete-data log-likelihood is of the form

$$l = \sum_{i=1}^{N} C_i \; l_1^* + (1 - C_i) \; l_2^* \; , \tag{3.4}$$

where

$$l_1^* = \sum_{i=1}^{N} W_i \; \log \left(\frac{1}{\sqrt{(2\pi\sigma_1^2)}} \; \exp\{\frac{-1}{2\pi\sigma_1^2} \; (x_i - \mu_1)^2\} \right) + (1 - W_i) \; \log \left(\frac{1}{\sqrt{(2\pi\sigma_1^2)}} \; \exp\{\frac{-1}{2\pi\sigma_1^2} \; (x_i^* - \mu_1)^2\} \right)$$

and

$$l_2^* = \sum_{i=1}^{N} W_i \; \log \left(\frac{1}{\sqrt{(2\pi\sigma_2^2)}} \; \exp\{\frac{-1}{2\pi\sigma_2^2} \; (y_i - \mu_2)^2\} \right) + (1 - W_i) \; \log \left(\frac{1}{\sqrt{(2\pi\sigma_2^2)}} \; \exp\{\frac{-1}{2\pi\sigma_2^2} \; (y_i^* - \mu_2)^2\} \right)$$

So the log-likelihood for the i-th observation, l_i, can be written as

$$\begin{aligned} l_i = C_i & \left\{ W_i \; \left[-\log(\sigma_1) - \frac{1}{2\pi\sigma_1^2} \; (x_i - \mu_1)^2 \right] + (1 - W_i) \; \left[-\log(\sigma_1) - \frac{1}{2\pi\sigma_1^2} \; (x_i^* - \mu_1)^2 \right] \right\} \\ & + (1 - C_i) \left\{ W_i \; \left[-\log(\sigma_2) - \frac{1}{2\pi\sigma_2^2} \; (y_i - \mu_2)^2 \right] + (1 - W_i) \; \left[-\log(\sigma_2) - \frac{1}{2\pi\sigma_2^2} \; (y_i^* - \mu_2)^2 \right] \right\} \end{aligned} \tag{3.5}$$

Notice that $L = \sum_{i=1}^{N} l_i + c$, where c is a normalizing constant depending only on the data.

3.4.2 Parameter estimation

As suggested in Aitkin et al. (2009), we implement an EM algorithm to estimate the censored survival times. The EM algorithm introduced by Dempster et al. (1977) is a general technique for finding maximum likelihood estimates for parametric models when the data are not fully observed, and where maximum likelihood estimation is difficult due to the presence of missing data. In general, it consists of iteration over two steps: the E-step, which involves finding the conditional expectation of the complete-data log-likelihood over the unobserved/latent variables given the current parameter estimates and the observed data, and the M-step, in which the expected complete-data log-likelihood is maximized to give updated parameter estimates. In this section, we implement a new EM algorithm to estimate the parameters of the model in the E-step using the first and the second moments of the truncated normal distribution. The higher order moments will then be used to obtain the standard error estimates.

E-step Since in the normal model the likelihood L is linear in x_i, y_i, x_i^2 and y_i^2, the E-step requires obtaining the conditional expected values of x_i, y_i, x_i^2 and y_i^2 given the current estimates for the censored observations (x_i^*, y_i^*, $x_i^{2^*}$ and $y_i^{2^*}$), the observed data and the current parameter estimates. These values are then replaced in the L, to obtain the conditional expected values of the L. This updating of the individual values of the sufficient statistics simplifies the E-step.

M-step Involves the maximization of the complete-data log-likelihood L by solving the likelihood equations to give new parameter estimates using the conditional expectations obtained in the E-step as real data.

The iterative process stops when estimates do not change much from one step to the next, i.e. having a reasonably small deviations in the norm metrics which are given below in (3.12).

Thus, in the **E-step**, we construct the expected observations from the current parameter estimates at stage ν, $\boldsymbol{\theta}^{(\nu)}$. This requires the results from the Proposition on moments of a right censored normal distribution given in Section 3.2.

$$\begin{aligned}
\tilde{x_i} &= E(X_i^*|X_i^* > z_i, \boldsymbol{\theta}^{(\nu)}) \\
&= C_i W_i z_i + C_i \ (1 - W_i) \left[{\mu_1}^{(\nu)} + \sigma_1 \ h(\frac{z_i - \mu_1^{(\nu)}}{\sigma_1^{(\nu)}}) \right] \qquad (3.6) \\
\tilde{y_i} &= E(Y_i^*|Y_i^* > z_i, \boldsymbol{\theta}^{(\nu)}) \\
&= (1 - C_i) W_i z_i + (1 - C_i) \ (1 - W_i) \left[\mu_2^{(\nu)} + \sigma_2^{(\nu)} \ h(\frac{z_i - \mu_2^{(\nu)}}{\sigma_2^{(\nu)}}) \right] \qquad (3.7) \\
\tilde{x_i^2} &= E(X_i^{2^*}|{X_i}^* > z_i, \boldsymbol{\theta}^{(\nu)}) \\
&= C_i W_i z_i^2 + C_i (1 - W_i) \left[{\mu_1^{(\nu)}}^2 + {\sigma_1^{(\nu)}}^2 + \sigma_1^{(\nu)} (\mu_1^{(\nu)} + z_i) \ h(\frac{z_i - \mu_1^{(\nu)}}{\sigma_1^{(\nu)}}) \right] \qquad (3.8)
\end{aligned}$$

$$\begin{aligned}\tilde{y_i^2} &= E(Y_i^{2*}|Y_i^* > z_i, \boldsymbol{\theta}^{(\nu)}) \\ &= (1-C_i)W_i z_i^2 + (1-C_i)(1-W_i)\left[{\mu_2^{(\nu)}}^2 + {\sigma_2^{(\nu)}}^2 + \sigma_2^{(\nu)}(\mu_2^{(\nu)} + z_i)h(\frac{z_i - \mu_2^{(\nu)}}{\sigma_2^{(\nu)}})\right]\end{aligned} \tag{3.9}$$

It is worth mentioning that according to the results in Section 3.2, $E(X_i^{2*}|X_i^* > z_i) \neq E^2(X_i^*|X_i^* > z_i)$, $i = 1, 2, \ldots, N$, necessarily (and the same holds for Y_i^{2*}). So the estimator for σ^2 represented in (Wolynetz, 1979, Page 186) is in error.

The **M-step** gives the estimate updates as follows:

$$\mu_1^{(\nu+1)} = \frac{\sum_{i=1}^N \tilde{x}_i}{\sum_{i=1}^N C_i} \quad , \quad \sigma_1^{(\nu+1)} = \frac{\sqrt{\tilde{x_i^2} - 2{\mu_1}^{(\nu)}\tilde{x}_i + \sum_{i=1}^N C_i \ {\mu_1^{(\nu)}}^2}}{\sum_{i=1}^N C_i} \tag{3.10}$$

$$\mu_2^{(\nu+1)} = \frac{\sum_{i=1}^N \tilde{y}_i}{\sum_{i=1}^N (1-C_i)} \quad , \quad \sigma_2^{(\nu+1)} = \frac{\sqrt{\tilde{y_i^2} - 2\mu_2^{(\nu)}\tilde{y}_i + \sum_{i=1}^N (1-C_i){\mu_2^{(\nu)}}^2}}{\sum_{i=1}^N (1-C_i)} \tag{3.11}$$

Given the parameter estimates obtained in this step, the expected values of the censored data gets updated in the E-Step. We repeat this procedure until the criteria

$$\left|l(\Theta^\nu) - l(\Theta^{\nu-1})\right| < 10^{-5} \quad \text{and} \quad \left|\frac{\Theta^\nu - \Theta^{\nu-1}}{\Theta^\nu}\right| < 10^{-5} \tag{3.12}$$

hold, where ν is the iteration counter. Then Θ^ν is the MLE of the vector of parameters.

3.4.3 Estimation of standard errors

Louis (1982) presents a commonly used method for obtaining standard errors when the EM algorithm is used. Based on this method, the observed information matrix is computed from the last iteration of the EM procedure, when the parameter estimates have converged to give $\hat{\boldsymbol{\theta}} = (\hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2, \hat{\pi})$. In the remainder of this section we find the prerequisites for implementing the Louis method step-by-step, so we can then calculate the observed information matrix and consequently find the standard errors for the parameters based on the incomplete data.

According to the Louis method, one can find the inverse of the variance-covariance matrix from the following formula,

$$I_L = B - A + I_1 \tag{3.13}$$

where $B = \sum_{i=1}^{N} E(B_i)$ and $B_i(= B_i^{ij})$ is the matrix of the negative second derivatives of $\tilde{l}_i$ denoting the conditional expectation of the complete-data log-likelihood for the i-th observation given the current parameter estimates , $\boldsymbol{\theta}^{(\nu)}$, and the observed data. The matrix $A(= \sum_{i=1}^{N} a_{jk})$ is defined as $\sum_{i=1}^{N} E\left(S_i.S_i^T\right)$, and also I_1 is defined as $\sum_{i=1}^{N} E(S_i).E^T(S_i)$ with S_i as defined in (3.14). The estimated variance-covariance matrix is equal to I_L^{-1}.

The gradient vector S_i based on the complete-data log-likelihood (3.5) can be found as following:

$$S_i = \begin{bmatrix} \frac{\partial l_i}{\partial \mu_1} \\ \frac{\partial l_i}{\partial \sigma_1} \\ \frac{\partial l_i}{\partial \mu_2} \\ \frac{\partial l_i}{\partial \sigma_2} \end{bmatrix} \tag{3.14}$$

where

$$\begin{aligned}
\frac{\partial l_i}{\partial \mu_1} &= \frac{C_i}{\sigma_1^2}\{W_i(z_i - \mu_1) + (1 - W_i)(x_i^* - \mu_1)\} \\
\frac{\partial l_i}{\partial \sigma_1} &= C_i\left\{W_i\left[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(z_i - \mu_1)^2\right] + (1 - W_i)\left[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(x_i^* - \mu_1)^2\right]\right\} \\
\frac{\partial l_i}{\partial \mu_2} &= \frac{1 - C_i}{\sigma_2^2}\{W_i(z_i - \mu_2) + (1 - W_i)(y_i^* - \mu_2)\} \\
\frac{\partial l_i}{\partial \sigma_2} &= (1 - C_i)\left\{W_i\left[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}(z_i - \mu_2)^2\right] + (1 - W_i)\left[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}(y_i^* - \mu_2)^2\right]\right\}.
\end{aligned}$$

The matrix $B_i = B_i^{jk}$, $j, k = 1, \ldots, 4$, written below, is the negative of the second-order partial derivative matrix of (3.5) with respect to the parameters, where the notation A^{jk} is used for the element in the i-th row and j-th column of a matrix A.

So we have,

$$
\begin{aligned}
B_i^{11} &= -\frac{\partial^2 l_i}{\partial \mu_1^2} = \frac{C_i}{\sigma_1^2} \\
B_i^{12} &= B_i^{21} = -\frac{\partial^2 l_i}{\partial \mu_1 \partial \sigma_1} = -\frac{\partial^2 l_i}{\partial \sigma_1 \partial \mu_1} = \frac{2C_i}{\sigma_1^3}\big\{W_i(z_i-\mu_1)+(1-W_i)(x_i^*-\mu_1)\big\}
\end{aligned}
$$

$$
\begin{aligned}
B_i^{22} &= -\frac{\partial^2 l_i}{\partial \sigma_1^2} = -C_i\Big\{W_i\Big[\frac{1}{\sigma_1^2}-\frac{3}{\sigma_1^4}(z_i-\mu_1)^2\Big]+(1-W_i)\left[\frac{1}{\sigma_1^2}-\frac{3}{\sigma_1^4}(x_i^*-\mu_1)^2\right]\Big\} \\
B_i^{33} &= -\frac{\partial^2 l_i}{\partial \mu_2^2} = \frac{1-C_i}{\sigma_2^2} \\
B_i^{34} &= B_i^{43} = -\frac{\partial^2 l_i}{\partial \mu_2 \partial \sigma_2} = -\frac{\partial^2 l_i}{\partial \sigma_2 \partial \mu_2} = \frac{2(1-C_i)}{\sigma_2^3}\big\{W_i(z_i-\mu_2)+(1-W_i)(y_i^*-\mu_2)\big\} \\
B_i^{44} &= -\frac{\partial^2 l_i}{\partial \sigma_2^2} = -(1-C_i)\Big\{W_i\Big[\frac{1}{\sigma_2^2}-\frac{3}{\sigma_2^4}(z_i-\mu_2)^2\Big]+(1-W_i)\Big[\frac{1}{\sigma_2^2}-\frac{3}{\sigma_2^4}(y_i^*-\mu_2)^2\Big]\Big\},
\end{aligned}
$$

and other elements in B_i are equal to zero.

Now having the fact that regularity conditions hold, in order to find the Fisher Information (FI) matrix $B = \sum_{i=1}^{N} E_{\hat{\Theta}}(B_i)$, we first need to find the conditional expectations of B_i^{jk}, $j, k = 1, \ldots, 4$ as follows:

$$
\begin{aligned}
E(B_i^{11}) &= \frac{C_i}{\sigma_1^2} \\
E(B_i^{12}) &= E(B_i^{21}) = \frac{2C_i}{\sigma_1^3}\big\{W_i(z_i-\mu_1)+(1-W_i)(\tilde{x}_i-\mu_1)\big\} \\
E(B_i^{22}) &= -C_i\Big\{W_i\Big[\frac{1}{\sigma_1^2}-\frac{3}{\sigma_1^4}(z_i-\mu_1)^2\Big]+(1-W_i)\left[\frac{1}{\sigma_1^2}-\frac{3}{\sigma_1^4}(\tilde{x}_i-\mu_1)^2\right]\Big\} \\
E(B_i^{33}) &= \frac{1-C_i}{\sigma_2^2} \\
E(B_i^{34}) &= E(B_i^{43}) = \frac{2(1-C_i)}{\sigma_2^3}\big\{W_i(z_i-\mu_2)+(1-W_i)(\tilde{y}_i-\mu_2)\big\} \\
E(B_i^{44}) &= -(1-C_i)\Big\{W_i\Big[\frac{1}{\sigma_2^2}-\frac{3}{\sigma_2^4}(z_i-\mu_2)^2\Big]+(1-W_i)\Big[\frac{1}{\sigma_2^2}-\frac{3}{\sigma_2^4}(\tilde{y}_i-\mu_2)^2\Big]\Big\}
\end{aligned}
$$

The next step for finding the standard errors using the Louis method is finding the conditional expectation of the symmetric matrix $S_i.S_i^T$.

$$S_i.S_i^T = \begin{bmatrix} \left(\frac{\partial l_i}{\partial \mu_1}\right)^2 & \left(\frac{\partial l_i}{\partial \mu_1}\right)\left(\frac{\partial l_i}{\partial \sigma_1}\right) & 0 & 0 \\ & \left(\frac{\partial l_i}{\partial \sigma_1}\right)^2 & 0 & 0 \\ & & \left(\frac{\partial l_i}{\partial \mu_2}\right)^2 & \left(\frac{\partial l_i}{\partial \mu_2}\right)\left(\frac{\partial l_i}{\partial \sigma_2}\right) \\ & & & \left(\frac{\partial l_i}{\partial \sigma_2}\right)^2 \end{bmatrix} \tag{3.15}$$

Let $A(=\sum_{i=1}^{N} a_{jk}) = \sum_{i=1}^{N} E_{\hat{\Theta}}(S_i.S_i^T)$. So A is defined by its elements derived as follows.

$$\begin{aligned} a_{11} &= \frac{C_i^2}{\sigma_1^4}\left\{W_i^2(Z_i-\mu_1)^2 + (1-W_i)^2 E(X_i^*-\mu_1)^2\right\} \\ &= \frac{C_i^2}{\sigma_1^4}\left\{W_i^2(z_i-\mu_1)^2 + (1-W_i)^2 D_1\right\} \end{aligned}$$

where

$$D_1 = E[(X_i^*-\mu_1)^2 | X_i^* > z_i] = \tilde{x_i^2} - 2\mu_1\tilde{x}_i + {\mu_1}^2. \tag{3.16}$$

$\tilde{x}_i$ and $\tilde{x_i^2}$ are already defined in (3.6) and (3.8).

$$\begin{aligned} a_{12} =& \frac{C_i^2}{\sigma_1^2}\Big\{W_i^2(Z_i-\mu_1)\Big(-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(Z_i-\mu_1)^2\Big) + (1-W_i)^2 E\big[(X_i^*-\mu_1)(-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(X_i^*-\mu_1)^2\Big)\big]\Big\} \\ =& \frac{C_i^2}{\sigma_1^2}\left\{W_i^2(z_i-\mu_1)\Big(-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\Big) + (1-W_i)^2\left[\frac{-1}{\sigma_1}(\tilde{x}_i-\mu_1)+\frac{1}{\sigma_1^3}D_2\right]\right\} \end{aligned}$$

where

$$D_2 = E[(X_i^*-\mu_1)^3 | X_i^* > z_i] = \tilde{x_i^3} - \mu_1^3 - 3\tilde{x_i^2}\mu_1 + 3\mu_1^2\tilde{x}_i \tag{3.17}$$

and $\tilde{x_i^3}$, is defined as below (from the expression given in Section 3.2):

$$\tilde{x_i^3} = C_iW_iz_i^3 + C_i(1-W_i)\mu_1^3 + 3\sigma_1^2\mu_1 + [2\sigma_1^3 + \sigma_1(z_i^2+\mu_1^2+\mu_1 z_i)]h\Big(\frac{z_i-\mu_1}{\sigma_1}\Big). \tag{3.18}$$

Please note that in performing the above calculations, we made use of the fact that $W_i(1-W_i)=0$.

$$a_{22} = C_i^2\Big\{W_i^2\Big[-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(Z_i-\mu_1)^2\Big]^2+(1-W_i)^2\Big[\frac{1}{\sigma_1^2}+\frac{1}{\sigma_1^6}D_3-\frac{2}{\sigma_1^4}D_1\Big]\Big\}$$

where

$$D_3 = E[(X_i^*-\mu_1)^4|X_i^*>z_i] = \tilde{x_i^4}-4\mu_1\tilde{x_i^3}+6\mu_1\tilde{x_i^2}-4\mu_1^3\tilde{x}_i+\mu_1^4 \tag{3.19}$$

and

$$\tilde{x_i^4} = \mu_1^4+3\sigma_1^4+6\mu_1^2\sigma_1^2+\{5\sigma_1^3\mu_1+\mu_1^3\sigma_1+3\sigma_1^3z_i+\mu_1^2\sigma_1z_i+\mu_1\sigma_1z_i^2+\sigma_1z_i^3\}\ h(\frac{z_i-\mu_1}{\sigma_1}). \tag{3.20}$$

$$a_{33} = \frac{(1-C_i)^2}{\sigma_2^4}\left\{W_i^2(z_i-\mu_2)^2+(1-W_i)^2D_4\right\}$$

where

$$D_4 = E[(Y_i^*-\mu_2)^2|Y_i^*>z_i] = \tilde{y_i^2}-2\mu_2\tilde{y}_i+\mu_2^2. \tag{3.21}$$

$\tilde{y}_i$ and $\tilde{y_i^2}$ are already defined in (3.7) and (3.9).

$$\begin{aligned}a_{34} =&\frac{(1-C_i)^2}{\sigma_2^2}\Big\{W_i^2(z_i-\mu_2)\Big[-\frac{1}{\sigma_2}+\frac{1}{\sigma_2^3}(Z_i-\mu_2)\Big]\\&+(1-W_i)^2\Big[\frac{-1}{\sigma_2^2}(\tilde{y}_i-\mu_2)+\frac{1}{\sigma_2^3}D_5\Big]\Big\}\end{aligned}$$

where

$$D_5 = E[(Y_i^*-\mu_2)^3|Y_i^*>z_i] = \tilde{y_i^3}-\mu_2^3-3\tilde{y_i^2}\mu_2+3\mu_2^2\tilde{y}_i \tag{3.22}$$

and

$$\begin{aligned}\tilde{y_i^3} =&(1-C_i)W_iz_i^3 \\ &+(1-C_i)(1-W_i)\mu_2^3+3\sigma_2^2\mu_2+[2\sigma_2^3+\sigma_2(z_i^2+\mu_2^2+\mu_2z_i)]h\Big(\frac{z_i-\mu_2}{\sigma_2}\Big).\end{aligned} \tag{3.23}$$

$$a_{44} = (1-C_i)^2\Big\{W_i^2\Big[-\frac{1}{\sigma_2}+\frac{1}{\sigma_2^3}(Z_i-\mu_2)^2\Big]^2+(1-W_i)^2\Big[\frac{1}{\sigma_2^2}+\frac{1}{\sigma_2^6}D_6-\frac{2}{\sigma_2^4}D_4\Big]\Big\}$$

where

$$D_6 = E[(Y_i^* - \mu_2)^4 | Y_i^* > z_i] = \tilde{y_i^4} - 4\mu_2\tilde{y_i^3} + 6\mu_2\tilde{y_i^2} - 4\mu_2^3\tilde{y}_i + \mu_2^4. \tag{3.24}$$

and

$$\tilde{y_i^4} = (1-\hat{U}_i)W_i z_i^3 + (1-\hat{U}_i)(1-W_i)\{\mu_2^4 + 3\sigma_2^4 + 6\mu_2^2\sigma_2^2 + \{5\sigma_2^3\mu_2 + \mu_2^3\sigma_2 + 3\sigma_2^3 z_i$$
$$+ \mu_2^2\sigma_2 z_i + \mu_2\sigma_2 z_i^2 + \sigma_2 z_i^3\}\ h(\frac{z_i - \mu_2}{\sigma_2})\}. \tag{3.25}$$

The last step before finding the inverse of variance-covariance (Var-Cov) matrix based on the Louis method, is to calculate the expected gradient vector and then calculate the matrix $I_1 = \sum_{i=1}^{N} E_{\hat{\Theta}}(S_i)E_{\hat{\Theta}}^T(S_i)$. The matrix $E(S_i)$ is given by

$$E(S_i) = \begin{bmatrix} \frac{C_i}{\sigma_1^2}\big\{W_i(z_i-\mu_1) + (1-W_i)(\tilde{x}_i - \mu_1)\big\} \\ C_i\big\{W_i\big[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(z_i-\mu_1)^2\big] + (1-W_i)\big[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}D_1\big]\big\} \\ \frac{1-C_i}{\sigma_2^2}\big\{W_i(z_i-\mu_2) + (1-W_i)(\tilde{y}_i - \mu_2)\big\} \\ (1-C_i)\big\{W_i\big[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}(z_i-\mu_2)^2\big] + (1-W_i)\big[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}D_4\big]\big\} \end{bmatrix} \tag{3.26}$$

Therefore, according to equation (3.13) the Var-Cov matrix inverse can be obtained. Hence, the Var-Cov matrix is equal to I_L^{-1} and so the standard error of parameters can be found. See Section 3.7.4 for the numerical results.

3.5 Mixed normal life-time population

Finite mixture models have been frequently used to model data. In this part, we deal with the calculation of the MLE's of the means and the standard deviations of the mixed-normal distribution. Suppose that N independent observations $z_1, z_2, \ldots, z_N$ are taken from the two component mixture density $f(z) = \pi f_1(z) + (1-\pi)f_2(z)$ representing the life-times of the individuals under a life-testing experiment.

3.5.1 Model

The likelihood for such an experiment explained above can be defined by considering a latent indicator variable U_i, $i = 1, 2, \ldots, N$, as the component indicator for the observations. This allows us to write down the complete-data likelihood in the form

$$L = \prod_{i=1}^{N} \left\{f_1(z_i)\right\}^{U_i} \left\{f_2(z_i)\right\}^{1-U_i} \pi^{U_i}(1-\pi)^{1-U_i}.$$

Hence, the log-likelihood l can be written as:

$$l = \sum_{i=1}^{N} U_i \log\left(\frac{\pi}{\sqrt{2\pi\sigma_1^2}} \exp\{\frac{-1}{2\sigma_1^2}(z_i-\mu_1)^2\}\right) + \sum_{i=1}^{N}(1-U_i)\log\left(\frac{1-\pi}{\sqrt{2\pi\sigma_2^2}} \exp\{\frac{-1}{2\sigma_2^2}(z_i-\mu_2)^2\}\right). \quad (3.27)$$

3.5.2 Parameter estimation

We use the EM algorithm for the fitting of this mixture model. As explained before it involves two steps which are repeated iteratively until the criteria (3.12) hold for the vector of parameters $\Theta = (\mu_1, \sigma_1, \mu_2, \sigma_2, \pi)$.

Here the **E-step** involves replacing the missing variable U_i by its conditional expectation, given the data and current parameter estimates, noting the fact that the likelihood function (3.27) is linear in U_i. The estimator for U_i is given by

$$\begin{aligned} \hat{U}_i &= E(U_i|\boldsymbol{z}, \Theta) \\ &= P(U_i = 1|\boldsymbol{z}, \Theta) \\ &= \frac{\pi f_1(\mu_1, \sigma_1)}{\pi f_1(\mu_1, \sigma_1) + (1-\pi) f_2(\mu_2, \sigma_2)} \end{aligned} \quad (3.28)$$

which follows using the Bayes theorem.

In the **M-step** the complete-data log-likelihood gets maximized. The solutions to the likelihood equations, given the data and current parameter estimates can be written simply as below by finding the first derivative of (3.27) with respect to the parameters.

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N} \hat{U}_i z_i}{\sum_{i=1}^{N} \hat{U}_i} \quad , \quad \hat{\sigma}_1 = \left\{ \frac{\sum_{i=1}^{N} \hat{U}_i (z_i - \hat{\mu}_1)^2}{\sum_{i=1}^{N} \hat{U}_i} \right\}^{1/2} \quad (3.29)$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{N}(1-\hat{U}_i)z_i}{\sum_{i=1}^{N}(1-\hat{U}_i)} \qquad , \qquad \hat{\sigma}_2 = \left\{\frac{\sum_{i=1}^{N}(1-\hat{U}_i)(z_i-\hat{\mu}_2)^2}{\sum_{i=1}^{N}(1-\hat{U}_i)}\right\}^{1/2} \tag{3.30}$$

$$\hat{\pi} = \frac{\sum_{i=1}^{N}\hat{U}_i}{N} \tag{3.31}$$

3.5.3 Estimation of standard errors

In order to find the standard errors of the parameters, we need the log-likelihood for the i-th, $i = 1, 2, \ldots, N$, observation. This is given by

$$\begin{aligned} l_i = {} & U_i\left[-\log(\sigma_1) - \frac{1}{2\pi\sigma_1^2}\ (z_i-\mu_1)^2 + \log(\pi)\right] \\ & + (1-U_i)\left[-\log(\sigma_2) - \frac{1}{2\pi\sigma_2^2}\ (z_i-\mu_2)^2 + \log(1-\pi)\right] \end{aligned} \tag{3.32}$$

The gradient vector based on the complete-data log-likelihood (3.32) is equal to $S_i = (\frac{\partial l_i}{\partial \mu_1}, \frac{\partial l_i}{\partial \sigma_1}, \frac{\partial l_i}{\partial \mu_2}, \frac{\partial l_i}{\partial \sigma_2}, \frac{\partial l_i}{\partial \pi})^T$, where

$$\begin{aligned} \frac{\partial l_i}{\partial \mu_1} &= \frac{U_i}{\sigma_1^2}(z_i-\mu_1) \\ \frac{\partial l_i}{\partial \sigma_1} &= U_i\left[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right] \\ \frac{\partial l_i}{\partial \mu_2} &= \frac{1-U_i}{\sigma_2^2}(z_i-\mu_2) \\ \frac{\partial l_i}{\partial \sigma_2} &= (1-U_i)\left[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}(z_i-\mu_2)^2\right] \\ \frac{\partial l_i}{\partial \pi} &= \frac{U_i}{\pi} - \frac{1-U_i}{1-\pi} \end{aligned}$$

The matrix $B_i = B_i^{jk}$, $j, k = 1, \ldots, 4$, is negative the second derivative matrix of (3.32) with respect to the parameters. The non-zero elements are derived as follows.

$$\begin{aligned} B_i^{11} &= -\frac{\partial^2 l_i}{\partial \mu_1^2} = \frac{U_i}{\sigma_1^2} \\ B_i^{12} = B_i^{21} &= -\frac{\partial^2 l_i}{\partial \mu_1 \partial \sigma_1} = \frac{\partial^2 l_i}{\partial \sigma_1 \partial \mu_1} = -\frac{2U_i}{\sigma_1^3}(z_i-\mu_1) \end{aligned}$$

$$
\begin{aligned}
B_i^{22} &= -\frac{\partial^2 l_i}{\partial \sigma_1^2} = -U_i\Big[\frac{1}{\sigma_1^2} - \frac{3}{\sigma_1^4}(z_i-\mu_1)^2\Big] \\
B_i^{33} &= -\frac{\partial^2 l_i}{\partial \mu_2^2} = \frac{1-U_i}{\sigma_2^2} \\
B_i^{34} = B_i^{43} &= -\frac{\partial^2 l_i}{\partial \mu_2 \partial \sigma_2} = -\frac{\partial^2 l_i}{\partial \sigma_2 \partial \mu_2} = \frac{2(1-U_i)}{\sigma_2^3}(z_i-\mu_2) \\
B_i^{44} &= -\frac{\partial^2 l_i}{\partial \sigma_2^2} = -(1-U_i)\Big[\frac{1}{\sigma_2^2} - \frac{3}{\sigma_2^4}(z_i-\mu_2)^2\Big] \\
B_i^{55} &= -\frac{\partial^2 l_i}{\partial \pi^2} = \frac{U_i}{\pi^2} + \frac{1-U_i}{(1-\pi)^2}
\end{aligned}
$$

Now regarding the fact that B_i^{jk}, $j,k = 1,\ldots,N$, are linear functions in U_i one can easily find the full information matrix $B = \sum_{i=1}^N E_{\hat{\Theta}}(B_i)$ by replacing the missing variable U_i in B_i with its expected value given by (3.28).

We let $A(= \sum_{i=1}^N a_{jk}) = E_{\hat{\Theta}}(S_i.S_i^T)$ where

$$
S_i.S_i^T = \begin{bmatrix}
\left(\frac{\partial l_i}{\partial \mu_1}\right)^2 & \left(\frac{\partial l_i}{\partial \mu_1}\right)\left(\frac{\partial l_i}{\partial \sigma_1}\right) & 0 & 0 & \left(\frac{\partial l_i}{\partial \mu_1}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\
 & \left(\frac{\partial l_i}{\partial \sigma_1}\right)^2 & 0 & 0 & \left(\frac{\partial l_i}{\partial \sigma_1}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\
 & & \left(\frac{\partial l_i}{\partial \mu_2}\right)^2 & \left(\frac{\partial l_i}{\partial \mu_2}\right)\left(\frac{\partial l_i}{\partial \sigma_2}\right) & \left(\frac{\partial l_i}{\partial \mu_2}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\
 & & & \left(\frac{\partial l_i}{\partial \sigma_2}\right)^2 & \left(\frac{\partial l_i}{\partial \sigma_2}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\
 & & & & \left(\frac{\partial l_i}{\partial \pi}\right)^2
\end{bmatrix}. \qquad (3.33)
$$

and A is derived as follows.

$$
\begin{aligned}
a_{11} &= E(U_i^2)\frac{(z_i-\mu_1)^2}{\sigma_1^4} \\
&= \frac{(z_i-\mu_1)^2}{\sigma_1^4}\hat{U}_i \qquad\qquad \text{since } U_i \text{ is an indicator.} \\
a_{12} &= E(U_i^2)\left[\frac{1}{\sigma_1^2}\left\{(z_i-\mu_1)\left(\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right)\right\}\right] \\
&= \hat{U}_i\left[\frac{1}{\sigma_1^2}\left\{(z_i-\mu_1)\left(\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right)\right\}\right]
\end{aligned}
$$

$$
\begin{aligned}
a_{22} &= E(U_i^2)\left[\frac{-1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right]^2 = \hat{U}_i\left[\frac{-1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right]^2 \\
a_{33} &= \frac{(z_i-\mu_2)^2}{\sigma_2^4}E(1-U_i) = \frac{(z_i-\mu_2)^2}{\sigma_2^4}(1-\hat{U}_i) \\
a_{44} &= E(1-U_i)^2\left[\frac{-1}{\sigma_2}+\frac{1}{\sigma_2^3}(z_i-\mu_2)^2\right]^2 = (1-\hat{U}_i)\left[\frac{-1}{\sigma_2}+\frac{1}{\sigma_2^3}(z_i-\mu_2)^2\right]^2 \\
a_{55} &= \frac{E(U_i^2)}{\pi^2}+\frac{E(1-U_i)^2}{(1-\pi)^2}-2\,\frac{E\left[U_i(1-U_i)\right]}{\pi(1-\pi)} \\
&= \frac{\hat{U}_i}{\pi^2}+\frac{1-\hat{U}_i}{(1-\pi)^2} \qquad\qquad \text{since } U_i(1-U_i)=0 \\
a_{15} &= \frac{E(U_i^2)}{\pi\sigma_1^2}(z_i-\mu_1) = \frac{\hat{U}_i}{\pi\sigma_1^2}(z_i-\mu_1) \\
a_{25} &= \frac{E(U_i^2)}{\pi}\left(\frac{-1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right) = \frac{\hat{U}_i}{\pi}\left(\frac{-1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right) \\
a_{35} &= -\frac{E(1-U_i)^2}{(1-\pi)\sigma_2^2}(z_i-\mu_2) = \frac{-(1-\hat{U}_i)}{(1-\pi)\sigma_2^2}(z_i-\mu_2)
\end{aligned}
$$

Now similar to Section 3.4.3, we find $E(S_i)$ in the following form. Once it is obtained, the Var-Cov matrix I_L can be found straightforward using the relation between the three matrices B, A and I_1, i.e. $I_L = B - A + I_1$.

$$
E(S_i) = \begin{bmatrix}
\frac{\hat{U}_i}{\sigma_1^2}(z_i-\mu_1) \\
\hat{U}_i\left[\frac{-1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right] \\
\frac{1-\hat{U}_i}{\sigma_2^2}(z_i-\mu_2) \\
(1-\hat{U}_i)\left[\frac{-1}{\sigma_2}+\frac{1}{\sigma_2^3}(z_i-\mu_2)^2\right] \\
\frac{\hat{U}_i}{\pi}-\frac{1-\hat{U}_i}{1-\pi}
\end{bmatrix} \tag{3.34}
$$

See the numerical results in Section (3.7.4)

For the sake of making sure the estimated mixing proportion remains within the interval $(0,1)$, we use the logit link when maximizing the log-likelihood using the `optim` function in `R`. As the logit link is defined as the log of the odds ratio, i.e. $\log[\pi/(1-\pi)]$, the estimate for π and its standard error obtained in this way should be transformed back to the original scale. The back-transformed variance can be found by using the delta method (see Powell, 2007).

In order to do so, we model π on the logit scale as a realization of a normal random variable with mean μ and variance σ^2 as follows:

$$logit(\pi_i; t) = \mu + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2).$$

Hence, according to (2.13) we have

$$\widehat{\sigma^2_\pi} \cong \hat{\sigma}^2 \hat{\pi}^2 (1 - \hat{\pi})^2, \tag{3.35}$$

where $\hat{\pi} = \dfrac{e^{\hat{\mu}}}{1 + e^{\hat{\mu}}}$ and σ^2_π is the variance on the back-transformed scale. The estimate for π can be transformed back to the original scale very easily.

3.6 Mixed (progressive)/Type-II censored data

Here, we consider the analysis of lifetime data from Type-II and progressive Type-II censoring schemes for a heterogeneous population modelled by a two-component normal mixture to mark the benefits of progressive censoring regime. The approach can be extended to other mixture distributions and to more than two components, although here the use of just two components is motivated by the notion of susceptible and non-susceptible individuals, as in Kuo and Peng (2000).

Suppose now that we have N items from the finite mixture introduced in Section 3.5 but the data are subjected to progressive Type-II or Type-II censoring. So the censored version of the data (observed data) under Type-II censoring is of the form $Z_1 \leq \ldots \leq Z_N$ where $Z_1 \leq \ldots \leq Z_r = Z_{r+1} = Z_{r+2} = \ldots = Z_N$. That is we now assume that observations $z_1, z_2, \ldots, z_N$ are N independent observations representing the life-times of the individuals under a life-testing experiment from the following two-component mixture density

$$f(\boldsymbol{z}) = \pi f_1(\boldsymbol{z}|\mu_1, \sigma_1) + (1 - \pi) f_2(\boldsymbol{z}|\mu_2, \sigma_2) \tag{3.36}$$

where $\boldsymbol{z} = (z_1, \ldots, z_N)$ are mixture observations from two groups $\boldsymbol{X}$ and $\boldsymbol{Y}$; such that those from the first group follow $f_1(\cdot|\mu_1, \sigma_1)$ and those from the other follow $f_2(\cdot|\mu_2, \sigma_2)$. Density functions $f_1(\cdot|\mu_1, \sigma_1)$ and $f_2(\cdot|\mu_2, \sigma_2)$ are assumed to be $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ respectively. We denote them by $f_1(\cdot)$ and $f_2(\cdot)$ respectively hereafter unless the parameter specification is needed.

The observations $z_1, z_2, \ldots, z_N$ are again assumed to be subject to progressive Type-II censoring, i.e., we assume the experiment is continued until observing a prefixed number of failures, r, and immediately after each failure some live units are removed from the experiment according to a censoring plan $\mathcal{R} = (R_1, R_2, \ldots, R_r)$. Based on this censoring scheme, R_i units are withdrawn after the i-th failure such that

$$R_i \sim Bin\left(N - i - \sum_{j=1}^{i-1} R_j, p\right) \quad \text{and} \quad R_r = N - r - \sum_{j=1}^{r-1} R_j.$$

It is worth mentioning that p is the parameter controlling the desired amount of censoring at each progressive censoring stage. Hence, for large values of r, the value of p cannot be chosen large as there will be only a few items that are allowed to be removed from the experiment and the condition $N - i \geq \sum_{j=1}^{i-1} R_j$, $\forall i$, $i = 1, \ldots, r$, must hold. Other strategies that could be considered include (i) a fixed $\mathcal{R}$ plan rather than random, (ii) depending p on i, censoring stage, for example through considering $\{p_i\}$ a decreasing sequence.

Notice that with this form of data, we could form a likelihood from the mixture density and a mixture survival function, to account for censored observations, and use optimization (such as `optim` function in `R`) to fit the model. However, instead, we are going to take advantage of the EM algorithm which can handle mixture components estimation and also can be used to estimate the censored observations as in Aitkin et al. (2009).

For each individual we observe either the failure or censoring time. For the sake of simplicity in writing the likelihood function, we assume censored observations are independently distributed from the observed failures, to be able to multiply their densities when forming the likelihood. The observed data are shown with the notation $\mathcal{O}$ hereafter.

To develop the likelihood function, we restate the problem as an incomplete-data problem by defining a latent variable for the missing information about the component membership of the observations. We write U_i, $i = 1, \ldots, N$, as the component indicator for the i-th observation, z_i, as follows

$$U_i = \begin{cases} 1 & i\text{-th observation} \in \text{group } 1 \\ & \\ 0 & i\text{-th observation} \in \text{group } 2 \end{cases} \quad i = 1, 2, \ldots, N. \tag{3.37}$$

Each observed failure has a contribution of the form f_j, $j = 1, 2$, to the likelihood if it comes from the j-th component. Denoting the unobserved censored observation associated with z_i by $z_i^* = U_i x_i^* + (1 - U_i) y_i^*$, its contributions to the complete-data log-likelihood are $f_1(z_i^*)$ and $f_2(z_i^*)$ if from the first and second component respectively. Therefore, the censored observations have contributions of the form $f_1(x_i^*)$ if it is of $\boldsymbol{X}$ type and of the form $f_1(y_i^*)$ if it is of $\boldsymbol{Y}$ type, where x_i^* and y_i^* denote the unobserved true failure times from the first and the second mixture components respectively.

The complete-data likelihood function is then given as follows

$$L(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x^*}, \boldsymbol{y^*}, \boldsymbol{U}) \propto \prod_{i=1}^{N} \left\{f_1(z_i)\right\}^{W_i U_i} \ \left\{f_2(z_i)\right\}^{W_i(1-U_i)} \ \left\{f_1\left(x_i^*\right)\right\}^{(1-W_i)U_i} \\ \times \left\{f_2\left(y_i^*\right)\right\}^{(1-W_i)(1-U_i)} \pi^{U_i}(1-\pi)^{(1-U_i)} \tag{3.38}$$

where $\boldsymbol{U} = (U_1, \ldots, U_N)$, W_i, $i = 1, \ldots, N$, is the censoring indicator and is 1 if an event is observed for the i-th observation, 0 if not observed, and $\boldsymbol{\theta} = (\mu_1, \sigma_1, \mu_2, \sigma_2, \pi)^T$ is the vector of unknown parameters.

Hence, the complete-data log-likelihood contribution for the i-th observation from a heterogeneous population with a two component mixture distribution can be written as follows:

$$l_i = c + W_i\, U_i\, \log(f_1(z_i)) + W_i\,(1 - U_i)\, \log(f_2(z_i)) + (1 - W_i)\, U_i \log(f_1\left(x_i^*\right)) \\ + (1 - W_i)(1 - U_i) \log(f_2\left(y_i^*\right)) + U_i \log(\pi) + (1 - U_i)\, \log(1 - \pi) \tag{3.39}$$

Therefore, for a two component normal mixture population we have:

$$\begin{aligned}
l_i =& c + W_i\ U_i \left[-\log(\sigma_1) - \frac{1}{2\sigma_1}(z_i - \mu_1)^2 \right] + (1 - W_i)\ U_i \Bigg[-\log(\sigma_1) \\
&- \frac{1}{2\sigma_1}\left(x_i^* - \mu_1\right)^2 \Bigg] + (1 - W_i) U_i \left[-\log(\sigma_2) - \frac{1}{2\sigma_2}\left(z_i - \mu_2\right)^2 \right] \\
&+ (1 - W_i)\ (1 - U_i) \left[-\log(\sigma_2) - \frac{1}{2\sigma_2}\left(y_i^* - \mu_2\right)^2 \right] + U_i\ \log(\pi) \\
&+ (1 - U_i)\ \log(1 - \pi)
\end{aligned} \tag{3.40}$$

and hence the complete-data log-likelihood is $l(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^{*}, \boldsymbol{y}^{*}, \boldsymbol{U}) = \sum_{i=1}^{N} l_i$ where c in (3.40) is a normalizing constant.

Note: Let $\overline{F_j}(z_i), j = 1, 2$, be the survival functions. Latent observations censored at z_i, have the contributions of the form $\overline{F_1}(z_i)$ or $\overline{F_2}(z_i)$ to the observed likelihood function assuming they are from the first or second components respectively. Therefore the observed likelihood function can be written as below.

$$L(\boldsymbol{\theta}|\mathcal{O}) = \prod_{i=1}^{N} \left\{\pi f_1(z_i) + (1 - \pi) f_2(z_i)\right\}^{W_i} \left\{\pi \overline{F_1}(z_i) + (1 - \pi)\overline{F_2}(z_i)\right\}^{1 - W_i}$$

Remark: For the case of an homogeneous population, considering a single density function $f(\cdot)$ and distribution function $F(\cdot)$ for z_i, $i = 1, \ldots, N$, the observed likelihood function is

$$\begin{aligned}
L(\mu, \sigma|\mathcal{O}) &= \prod_{i=1}^{N} f(z_i|\mu, \sigma)^{W_i} \left\{\overline{F}(z_i|\mu, \sigma)\right\}^{1 - W_i} \\
&\propto \prod_{i=1}^{N} \left[\frac{1}{\sigma} \mathrm{e}^{-\frac{1}{2\sigma^2}(z_i - \mu)^2}\right]^{W_i} \left[\int_{z_i}^{+\infty} \frac{1}{\sigma} \mathrm{e}^{-\frac{1}{2\sigma^2}(z_i - \mu)^2}\right]^{1 - W_i},
\end{aligned} \tag{3.41}$$

which can be maximized for μ and σ numerically using the optimization methods in `optim` function in `R`.

3.6.1 Parameter estimation

Here we consider estimating the unknown parameters in the proposed likelihood (3.40) using an EM algorithm to deal with the two forms of missing data: the censored survival times and the mixture component labels. Suppose the EM algorithm is at the ν-th iteration. First, in the E-step we obtain the mixture component indicator estimates given the current parameter estimates $\boldsymbol{\theta}^{(\nu)}$ and the observed data $\mathcal{O}$. Next, in order to re-estimate the unknown parameters, we nest a new EM algorithm in the M-step of the initial EM algorithm to deal with the estimation of the missing censored survival times and then build the maximum likelihood score equations.

First EM algorithm

As the partially observed complete-data log-likelihood function $l(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^{*}, \boldsymbol{y}^{*})$, is linear in U_i, to maximize it at the ν-th iteration of the EM, we first estimate the unobserved mixture component indicator U_i, $i = 1, \ldots, N$, in the E-step of the EM, given the current parameter estimate $\boldsymbol{\theta}^{(\nu)} = (\mu_1^{(\nu)}, \sigma_1^{(\nu)}, \mu_2^{(\nu)}, \sigma_2^{(\nu)}, \pi^{(\nu)})$ and the observed data $\mathcal{O}$. This is estimated by

$$
\begin{aligned}
\tilde{U}_i =& E\left(U_i|\mathcal{O};\boldsymbol{\theta}^{(\nu)}\right) \\
=& P\left(U_i = 1|\mathcal{O};\boldsymbol{\theta}^{(\nu)}\right), \quad i = 1, 2, \ldots, N \\
=& W_i \left\{ \frac{\pi^{(\nu)} f_1\left(z_i|\mu_1^{(\nu)}, \sigma_1^{(\nu)}\right)}{\pi^{(\nu)} f_1\left(z_i|\mu_1^{(\nu)}, \sigma_1^{(\nu)}\right) + (1-\pi^{(\nu)}) f_2\left(z_i|\mu_2^{(\nu)}, \sigma_2^{(\nu)}\right)} \right\} \\
&+ (1 - W_i) \left\{ \frac{\pi^{(\nu)} \overline{F_1}\left(z_i|\mu_1^{(\nu)}, \sigma_1^{(\nu)}\right)}{\pi^{(\nu)} \overline{F_1}\left(z_i|\mu_1^{(\nu)}, \sigma_1^{(\nu)}\right) + (1-\pi^{(\nu)}) \overline{F_2}\left(z_i|\mu_2^{(\nu)}, \sigma_2^{(\nu)}\right)} \right\}, \qquad (3.42)
\end{aligned}
$$

using Bayes theorem, where $\overline{F_1}$ and $\overline{F_2}$ are survival functions associated with f_1 and f_2.

Now using the above relation and equations (3.39) and (3.40), the conditional expectation of the complete-data log-likelihood for the i-th observation, denoted by $\tilde{l}_i$, given the current parameter estimate $\boldsymbol{\theta}^{(\nu)}$ and the observed data, is readily obtained as

$$
\begin{aligned}
\tilde{l}_i =& c + W_i \tilde{U}_i \log\left(f_1(z_i)\right) + W_i \left(1 - \tilde{U}_i\right) \log(f_2(z_i)) + (1 - W_i) \tilde{U}_i \log\left(f_1\left(x_i^{*}\right)\right) \\
&+ (1 - W_i)\left(1 - \tilde{U}_i\right) \log\left(f_2\left(y_i^{*}\right)\right) + \tilde{U}_i \log\left(\pi^{(\nu)}\right) + \left(1 - \tilde{U}_i\right) \log\left(1 - \pi^{(\nu)}\right)
\end{aligned}
$$

$$
\begin{aligned}
=&c + W_i\tilde{U}_i\left[-\log\left(\sigma_1^{(\nu)}\right) - \frac{1}{2\sigma_1^{(\nu)}}\left(z_i - \mu_1^{(\nu)}\right)^2\right] + (1-W_i)\tilde{U}_i\left[-\log\left(\sigma_1^{(\nu)}\right)\right.\\
&\left.- \frac{1}{2\sigma_1^{(\nu)}}\left(x_i^* - \mu_1^{(\nu)}\right)^2\right] + (1-W_i)\tilde{U}_i\left[-\log\left(\sigma_2^{(\nu)}\right) - \frac{1}{2\sigma_2^{(\nu)}}(z_i - \mu_2^{(\nu)})^2\right]\\
&+ (1-W_i)(1-\tilde{U}_i)\left[-\log\left(\sigma_2^{(\nu)}\right) - \frac{1}{2\sigma_2^{(\nu)}}\left(y_i^* - \mu_2^{(\nu)}\right)^2\right] + \tilde{U}_i\log\left(\pi^{(\nu)}\right)\\
&+ \left(1-\tilde{U}_i\right)\log\left(1-\pi^{(\nu)}\right).
\end{aligned}
\tag{3.43}
$$

In the M-step, the parameters are updated by maximization of the $l(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^*, \boldsymbol{y}^*, \tilde{\boldsymbol{U}}) = \sum_{i=1}^{N}\tilde{l}_i$ with respect to $\boldsymbol{\theta}$. For this purpose we first need to deal with the estimation of the censored information. Therefore, we nest a second EM algorithm here in the M-step to account for the censored data estimation, which requires the calculations of all sufficient statistics. This is explained in detail as below.

Second EM algorithm

In the E-step of this new EM algorithm, we find the expected value of missing information due to censoring. As the log-likelihood l_i given in (3.40) is linear in x_i^*, y_i^*, x_i^{2*} and y_i^{2*}, and $l(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^*, \boldsymbol{y}^*, \tilde{\boldsymbol{U}}) = \sum_{i=1}^{N} l_i$, the conditional expectation of $l(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^*, \boldsymbol{y}^*, \tilde{\boldsymbol{U}})$ in the E-step requires finding (i) the conditional expected values of the censored observations x_i^*, y_i^*, denoted by $\tilde{x}_i$ and $\tilde{y}_i$ (ii) the conditional expected values of x_i^{2*} and y_i^{2*} denoted by $\tilde{x_i^2}$ and $\tilde{y_i^2}$; given the current estimates for the censored observations, i.e. the censoring times, the observed data $\mathcal{O}$ and the current parameter estimates $\boldsymbol{\theta}^{(\nu)}$. To calculate these, we use the proposition in Section 3.2.

Using the results in Section 3.2 and the results from the first EM algorithm, given the current parameter estimates $\boldsymbol{\theta}^{(\boldsymbol{\nu})}$ and the observed data $\mathcal{O}$, we find the aforementioned conditional expectations for the missing elements in the complete-data log-likelihood as follows:

$$\begin{aligned}
\tilde{x_i} &= E\left(X_i^*|X_i^* > z_i, \mathcal{O}, \boldsymbol{\theta}^{(\nu)}\right) \\
&= \tilde{U}_i\ W_i\ z_i + \tilde{U}_i\ (1-W_i) \left[\mu_1^{(\nu)} + \sigma_1^{(\nu)}\ h\left(\frac{z_i-\mu_1^{(\nu)}}{\sigma_1^{(\nu)}}\right)\right] \\
\tilde{y_i} &= E\left(Y_i^*|Y_i^* > z_i, \mathcal{O}, \boldsymbol{\theta}^{(\nu)}\right) \\
&= \left(1-\tilde{U}_i\right) W_i\ z_i + \left(1-\tilde{U}_i\right)(1-W_i) \left[\mu_2^{(\nu)} + \sigma_2^{(\nu)}\ h\left(\frac{z_i-\mu_2^{(\nu)}}{\sigma_2^{(\nu)}}\right)\right] \\
\tilde{x_i^2} &= E\left(X_i^{2*}|X_i{}^* > z_i, \mathcal{O}, \boldsymbol{\theta}^{(\nu)}\right) \\
&= \tilde{U}_i\ W_i\ z_i^2 + \tilde{U}_i\ (1-W_i) \left[\mu_1^2 + {\sigma_1^{(\nu)}}^2 + \sigma_1^{(\nu)}\left(\mu_1^{(\nu)} + z_i\right)\ h\left(\frac{z_i-\mu_1^{(\nu)}}{\sigma_1^{(\nu)}}\right)\right] \\
\tilde{y_i^2} &= E\left(Y_i^{2*}|Y_i{}^* > z_i, \mathcal{O}, \boldsymbol{\theta}^{(\nu)}\right) \\
&= (1-\tilde{U}_i)\ W_i\ z_i^2 + \left(1-\tilde{U}_i\right)(1-W_i)\left[{\mu_2^{(\nu)}}^2 + {\sigma_2^{(\nu)}}^2 + \sigma_2^{(\nu)}\left(\mu_2^{(\nu)} + z_i\right)\right. \\
&\quad \left. \times\, h\left(\frac{z_i-\mu_2^{(\nu)}}{\sigma_2^{(\nu)}}\right)\right]
\end{aligned}$$

The M-step then involves the maximization of the expected complete-data log-likelihood $l(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x^*}, \boldsymbol{y^*}, \tilde{\boldsymbol{U}}) = \sum_{i=1}^N l_i$, given the current parameter estimates and the observed data.

The gradient vector $S_i = \dfrac{\partial l_i}{\partial \Theta}$ based on (3.43) can be obtained as follows:

$$S_i = \left(\frac{\partial l_i}{\partial \mu_1}, \frac{\partial l_i}{\partial \sigma_1}, \frac{\partial l_i}{\partial \mu_2}, \frac{\partial l_i}{\partial \sigma_2}, \frac{\partial l_i}{\partial \pi}\right)^T$$

$$= \begin{bmatrix}
\frac{U_i}{\sigma_1^2}\left\{W_i(z_i-\mu_1) + (1-W_i)(x_i^*-\mu_1)\right\} \\
U_i\left\{W_i\left[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(z_i-\mu_1)^2\right] + (1-W_i)\left[\frac{-1}{\sigma_1} + \frac{1}{\sigma_1^3}(x_i^*-\mu_1)^2\right]\right\} \\
\frac{1-U_i}{\sigma_2^2}\left\{W_i(z_i-\mu_2) + (1-W_i)(y_i^*-\mu_2)\right\} \\
(1-U_i)\left\{W_i\left[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}(z_i-\mu_2)^2\right] + (1-W_i)\left[\frac{-1}{\sigma_2} + \frac{1}{\sigma_2^3}(y_i^*-\mu_2)^2\right]\right\} \\
\frac{U_i}{\pi} - \frac{1-U_i}{1-\pi}
\end{bmatrix} \tag{3.44}$$

The parameters estimates are then given by the following formulae obtained by solving $\sum_{i=1}^{N} S_i = 0$ for the parameters. They involve the sufficient statistics that are updated based on the conditional expectations for the censored data, as given above.

$$\hat{\mu}_1^{(\nu+1)} = \frac{\sum_{i=1}^{N} \tilde{U}_i z_i}{\sum_{i=1}^{N} \tilde{U}_i} = \frac{\sum_{i=1}^{N} \tilde{U}_i \tilde{x}_i}{\sum_{i=1}^{N} \tilde{U}_i}, \tag{3.45}$$

$$\hat{\sigma}_1^{(\nu+1)} = \left[\frac{\sum_{i=1}^{N} \tilde{U}_i \left(z_i - \mu_1^{(\nu)} \right)^2}{\sum_{i=1}^{N} \tilde{U}_i} \right]^{\frac{1}{2}} = \left[\frac{\sum_{i=1}^{N} \tilde{U}_i \left(\tilde{x_i^2} - 2\mu_1^{(\nu)} \tilde{x}_i + \mu_1^{(\nu)2} \right)}{\sum_{i=1}^{N} \tilde{U}_i} \right]^{\frac{1}{2}}, \tag{3.46}$$

$$\hat{\mu}_2^{(\nu+1)} = \frac{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right) z_i}{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right)} = \frac{\sum_{i=1}^{N} (1 - \tilde{U}_i) \tilde{y}_i}{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right)}, \tag{3.47}$$

$$\begin{aligned} \hat{\sigma}_2^{(\nu+1)} &= \left[\frac{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right) \left(z_i - \mu_2^{(\nu)} \right)^2}{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right)} \right]^{\frac{1}{2}} \\ &= \left[\frac{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right) \left(\tilde{y_i^2} - 2\mu_1^{(\nu)} \tilde{y}_i + \mu_2^{(\nu)2} \right)}{\sum_{i=1}^{N} \left(1 - \tilde{U}_i\right)} \right]^{\frac{1}{2}}, \end{aligned} \tag{3.48}$$

$$\hat{\pi}^{(\nu+1)} = \frac{\sum_{i=1}^{N} \tilde{U}_i}{N}. \tag{3.49}$$

Given the parameter estimates obtained in this step, the expected values of missing data are then updated in the E-Step. We repeat these steps until the change in estimates are not significant from one step to the next, i.e. achieving convergence. We consider the Euclidean distance measure $|\boldsymbol{\theta}^{(\nu)} - \boldsymbol{\theta}^{(\nu-1)}| < 10^{-3}$ for accepting the parameter convergence at iteration ν. Also, to ensure the likelihood is converging we also check the criterion $\left| l^{(\nu)}(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^*, \boldsymbol{y}^*, \tilde{\boldsymbol{U}}) - l^{(\nu-1)}(\boldsymbol{\theta}|\mathcal{O}, \boldsymbol{W}, \boldsymbol{x}^*, \boldsymbol{y}^*, \tilde{\boldsymbol{U}}) \right| < 10^{-5}$. It is worth mentioning that it can be seen that the value of the log-likelihood gets larger at each iteration of the EM algorithm, as mentioned in Wu (1983).

3.6.2 Estimation of Standard Errors

The negative second derivative matrix B_i in Section 3.6.1 can be found as follows:

$$
\begin{aligned}
B_i^{11} &= -\frac{\partial^2 l_i}{\partial \mu_1^2} = \frac{U_i}{\sigma_1^2} \\
B_i^{12} = B_i^{21} &= -\frac{\partial^2 l_i}{\partial \mu_1 \partial \sigma_1} = -\frac{\partial^2 l_i}{\partial \sigma_1 \partial \mu_1} = \frac{2U_i}{\sigma_1^3}\big\{W_i(z_i-\mu_1)+(1-W_i)(x_i^*-\mu_1)\big\} \\
B_i^{22} &= -\frac{\partial^2 l_i}{\partial \sigma_1^2} = -U_i\Big\{W_i\Big[\frac{1}{\sigma_1^2}-\frac{3}{\sigma_1^4}(z_i-\mu_1)^2\Big]+(1-W_i)\left[\frac{1}{\sigma_1^2}-\frac{3}{\sigma_1^4}(x_i^*-\mu_1)^2\right]\Big\} \\
B_i^{33} &= -\frac{\partial^2 l_i}{\partial \mu_2^2} = \frac{1-U_i}{\sigma_2^2} \\
B_i^{34} = B_i^{43} &= -\frac{\partial^2 l_i}{\partial \mu_2 \partial \sigma_2} = -\frac{\partial^2 l_i}{\partial \sigma_2 \partial \mu_2} = \frac{2(1-U_i)}{\sigma_2^3}\big\{W_i(z_i-\mu_2)+(1-W_i)(y_i^*-\mu_2)\big\} \\
B_i^{44} &= -\frac{\partial^2 l_i}{\partial \sigma_2^2} = -(1-U_i)\Big\{W_i\Big[\frac{1}{\sigma_2^2}-\frac{3}{\sigma_2^4}(z_i-\mu_2)^2\Big]+(1-W_i)\Big[\frac{1}{\sigma_2^2}-\frac{3}{\sigma_2^4}(y_i^*-\mu_2)^2\Big]\Big\} \\
B_i^{55} &= -\frac{\partial^2 l_i}{\partial \pi^2} = \frac{U_i}{\pi^2}+\frac{1-U_i}{(1-\pi)^2}.
\end{aligned}
$$

Other elements in B_i are equal to zero.

The element-wise conditional expectation of the matrix B_i over $\boldsymbol{U_i}$, $\boldsymbol{x_i^*}$ and $\boldsymbol{y_i^*}$, given the observed data and the current parameter estimates, can be found as below.

$$
\begin{aligned}
E\left(B_i^{11}\right) &= E_{U_i}\left[B_i^{11}\right] = \frac{\hat{U}_i}{\hat{\sigma}_1^2}, \\
E\left(B_i^{12}\right) = E\left(B_i^{21}\right) &= E_{U_i}E_{X_i^*}\left[B_i^{12}|U_i\right] = \frac{2\tilde{U}_i}{\hat{\sigma}_1^3}\left\{W_i(z_i-\hat{\mu}_1)+(1-W_i)\left(\tilde{x}_i-\hat{\mu}_1\right)\right\}, \\
E\left(B_i^{22}\right) &= E_{U_i}E_{X_i^*}\left[B_i^{22}|U_i\right] = -\tilde{U}_i\left\{W_i\left[\frac{1}{\hat{\sigma}_1^2}-\frac{3}{\hat{\sigma}_1^4}(z_i-\hat{\mu}_1)^2\right]+(1-W_i)\left[\frac{1}{\hat{\sigma}_1^2}\right.\right. \\
&\qquad \left.\left. -\frac{3}{\hat{\sigma}_1^4}(\tilde{x}_i-\hat{\mu}_1)^2\right]\right\}, \\
E\left(B_i^{33}\right) &= E_{U_i}\left[B_i^{33}\right] = \frac{1-\tilde{U}_i}{\hat{\sigma}_2^2}, \\
E(B_i^{34}) = E\left(B_i^{43}\right) &= E_{U_i}E_{Y_i^*}\left[B_i^{34}|U_i\right] = \frac{2(1-\tilde{U}_i)}{\hat{\sigma}_2^3}\left\{W_i\left(z_i-\hat{\mu}_2\right)+(1-W_i)\left(\tilde{y}_i-\hat{\mu}_2\right)\right\}, \\
E\left(B_i^{44}\right) &= E_{U_i}E_{Y_i^*}\left[B_i^{44}|U_i\right] = -\left(1-\tilde{U}_i\right)\left\{W_i\left[\frac{1}{\hat{\sigma}_2^2}-\frac{3}{\hat{\sigma}_2^4}\left(z_i-\hat{\mu}_2\right)^2\right]+(1-W_i)\left[\frac{1}{\hat{\sigma}_2^2}\right.\right. \\
&\qquad \left.\left. -\frac{3}{\hat{\sigma}_2^4}(\tilde{y}_i-\hat{\mu}_2)^2\right]\right\},
\end{aligned}
$$

and

$$E\left(B_i^{55}\right) = E_{U_i}\left[B_i^{55}\right] = -\frac{\partial^2 l_i}{\partial \hat{\pi}^2} = \frac{\tilde{U}_i}{\hat{\pi}^2} + \frac{1-\tilde{U}_i}{\left(1-\hat{\pi}\right)^2}.$$

Hence,

$$E\left(B_i\right) = \left[\begin{array}{cc|cc|c} E\left(B_i^{11}\right) & E\left(B_i^{12}\right) & 0 & 0 & 0 \\ & E\left(B_i^{22}\right) & 0 & 0 & 0 \\ \hline & & E\left(B_i^{33}\right) & E\left(B_i^{34}\right) & 0 \\ & & & E\left(B_i^{44}\right) & 0 \\ \hline & & & & E\left(B_i^{55}\right) \end{array}\right]. \tag{3.50}$$

In order to derive matrix $A(=a_{ij}) = E\left(S_i.S_i^T\right)$, we first need to find matrix $S_i.S_i^T$ as follows.

$$S_i.S_i^T = \left[\begin{array}{cc|cc|c} \left(\frac{\partial l_i}{\partial \mu_1}\right)^2 & \left(\frac{\partial l_i}{\partial \mu_1}\right)\left(\frac{\partial l_i}{\partial \sigma_1}\right) & 0 & 0 & \left(\frac{\partial l_i}{\partial \mu_1}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\ & \left(\frac{\partial l_i}{\partial \sigma_1}\right)^2 & 0 & 0 & \left(\frac{\partial l_i}{\partial \sigma_1}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\ \hline & & \left(\frac{\partial l_i}{\partial \mu_2}\right)^2 & \left(\frac{\partial l_i}{\partial \mu_2}\right)\left(\frac{\partial l_i}{\partial \sigma_2}\right) & \left(\frac{\partial l_i}{\partial \mu_2}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\ & & & \left(\frac{\partial l_i}{\partial \sigma_2}\right)^2 & \left(\frac{\partial l_i}{\partial \sigma_2}\right)\left(\frac{\partial l_i}{\partial \pi}\right) \\ \hline & & & & \left(\frac{\partial l_i}{\partial \pi}\right)^2 \end{array}\right] \tag{3.51}$$

Notice that $E(B_i)$ and $S_i.S_i^T$ are both block diagonal symmetric matrices.

The matrix A is calculated as follows:

$$\begin{aligned} a_{11} &= \frac{\hat{U_i}^2}{\sigma_1^4}\left\{W_i^2(z_i-\mu_1)^2 + (1-W_i)^2 E(X_i^*-\mu_1)^2\right\} \\ &= \frac{\hat{U_i}^2}{\sigma_1^4}\left\{W_i^2(z_i-\mu_1)^2 + (1-W_i)^2 D_1\right\} \\ a_{12} &= \frac{\hat{U_i}^2}{\sigma_1^2}\Big\{W_i^2(z_i-\mu_1)\Big(-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\Big) + (1-W_i)^2 E\Big[(X_i^*-\mu_1)\Big(-\frac{1}{\sigma_1} \\ &\quad +\frac{1}{\sigma_1^3}(X_i^*-\mu_1)^2\Big)\Big]\Big\} \end{aligned}$$

$$= \frac{\hat{U_i}^2}{\sigma_1^2}\left\{W_i^2(z_i-\mu_1)\Big(-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\Big)+(1-W_i)^2\left[\frac{-1}{\sigma_1}(\tilde{x}_i-\mu_1)+\frac{1}{\sigma_1^3}D_2\right]\right\}$$

$$a_{22} = \hat{U_i}^2\Big\{W_i^2\Big[-\frac{1}{\sigma_1}+\frac{1}{\sigma_1^3}(z_i-\mu_1)^2\Big]^2+(1-W_i)^2\Big[\frac{1}{\sigma_1^2}+\frac{1}{\sigma_1^6}D_3-\frac{2}{\sigma_1^4}D_1\Big]\Big\}$$

$$a_{33} = \frac{(1-\hat{U}_i)^2}{\sigma_2^4}\Big\{W_i^2(z_i-\mu_2)^2+(1-W_i)^2D_4\Big\}$$

$$a_{34} = \frac{(1-\hat{U}_i)^2}{\sigma_2^2}\big\{W_i^2(z_i-\mu_2)\big[-\frac{1}{\sigma_2}+\frac{1}{\sigma_2^3}(z_i-\mu_2)\big]+(1-W_i)^2\big[\frac{-1}{\sigma_2^2}(\tilde{y}_i-\mu_2)+\frac{1}{\sigma_2^3}D_5\big]\big\}$$

$$a_{44} = (1-\hat{U}_i)^2\Big\{W_i^2\Big[-\frac{1}{\sigma_2}+\frac{1}{\sigma_2^3}(z_i-\mu_2)^2\Big]^2+(1-W_i)^2\Big[\frac{1}{\sigma_2^2}+\frac{1}{\sigma_2^6}D_6-\frac{2}{\sigma_2^4}D_4\Big]\Big\},$$

where

$$D_1 = E[(X_i^*-\mu_1)^2|X_i^*>z_i,\mathcal{O},\boldsymbol{\theta}^{(\nu)}] = \tilde{x_i^2}-2\mu_1\tilde{x}_i+\mu_1^2, \tag{3.52}$$

$$D_2 = E[(X_i^*-\mu_1)^3|X_i^*>z_i,\mathcal{O},\boldsymbol{\theta}^{(\nu)}] = \tilde{x_i^3}-\mu_1^3-3\tilde{x_i^2}\mu_1+3\mu_1^2\tilde{x}_i \quad \text{(see the proposition)} \tag{3.53}$$

$$D_3 = E(X_i^*-\mu_1)^4 = \tilde{x_i^4}-4\mu_1\tilde{x_i^3}+6\mu_1\tilde{x_i^2}-4\mu_1^3\tilde{x}_i+\mu_1^4, \tag{3.54}$$

$$D_4 = E[(Y_i^*-\mu_2)^2|Y_i^*>z_i,\mathcal{O},\boldsymbol{\theta}^{(\nu)}] = \tilde{y_i^2}-2\mu_2\tilde{y}_i+\mu_2^2, \tag{3.55}$$

$$D_5 = E[(Y_i^*-\mu_2)^3|Y_i^*>z_i,\mathcal{O},\boldsymbol{\theta}^{(\nu)}] = \tilde{y_i^3}-\mu_2^3-3\tilde{y_i^2}\mu_2+3\mu_2^2\tilde{y}_i, \tag{3.56}$$

$$D_6 = E[(Y_i^*-\mu_2)^4|Y_i^*>z_i,\mathcal{O},\boldsymbol{\theta}^{(\nu)}] = \tilde{y_i^4}-4\mu_2\tilde{y_i^3}+6\mu_2\tilde{y_i^2}-4\mu_2^3\tilde{y}_i+\mu_2^4, \tag{3.57}$$

and

$$\tilde{x_i^3} = \hat{U}_iW_iz_i^3+\hat{U}_i(1-W_i)\{\mu_1^3+3\sigma_1^2\mu_1+[2\sigma_1^3+\sigma_1({z_i}^2+\mu_1^2+\mu_1z_i)]h\big(\frac{z_i-\mu_1}{\sigma_1}\big)\}, \tag{3.58}$$

$$\begin{aligned}\tilde{x_i^4} &= \hat{U}_iW_iz_i^4+\hat{U}_i(1-W_i)\{\mu_1^4+3\sigma_1^4+6\mu_1^2\sigma_1^2+\{5\sigma_1^3\mu_1+\mu_1^3\sigma_1+3\sigma_1^3z_i+\mu_1^2\sigma_1z_i\\ &\quad+\mu_1\sigma_1{z_i}^2+\sigma_1{z_i}^3\}\ h(\frac{z_i-\mu_1}{\sigma_1})\},\end{aligned} \tag{3.59}$$

$$\begin{aligned}\tilde{y_i^3} &= (1-\hat{U}_i)W_iz_i^3+(1-\hat{U}_i)(1-W_i)\{\mu_2^3+3\sigma_2^2\mu_2\\ &\quad+[2\sigma_2^3+\sigma_2({z_i}^2+\mu_2^2+\mu_2z_i)]h\big(\frac{z_i-\mu_2}{\sigma_2}\big)\},\end{aligned} \tag{3.60}$$

$$\begin{aligned}\tilde{y_i^4} &= (1-\hat{U}_i)W_iz_i^4+(1-\hat{U}_i)(1-W_i)\{\mu_2^4+3\sigma_2^4+6\mu_2^2\sigma_2^2+\{5\sigma_2^3\mu_2+\mu_2^3\sigma_2+3\sigma_2^3z_i\\ &\quad+\mu_2^2\sigma_2z_i+\mu_2\sigma_2{z_i}^2+\sigma_2{z_i}^3\}\ h(\frac{z_i-\mu_2}{\sigma_2})\}.\end{aligned} \tag{3.61}$$

Notice that we made use of the fact that $W_i(1-W_i)=0$ in the above calculations.

We then need to find the expected value of the gradient vector (3.44), with respect to $\boldsymbol{U}$, $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$. This is obtained as below.

$$E(S_i) = \begin{bmatrix} \frac{\tilde{U}_i}{\sigma_1^2}\left\{W_i(z_i-\hat{\mu}_1)+(1-W_i)\left(\tilde{x_i}-\hat{\mu}_1\right)\right\} \\ \tilde{U}_i\left\{W_i\left[\frac{-1}{\hat{\sigma}_1}+\frac{1}{\hat{\sigma}_1^3}(z_i-\hat{\mu}_1)^2\right]+(1-W_i)\left[\frac{-1}{\hat{\sigma}_1}+\frac{1}{\hat{\sigma}_1^3}D_1\right]\right\} \\ \frac{1-\tilde{U}_i}{\hat{\sigma}_2^2}\left\{W_i\left(z_i-\hat{\mu}_2\right)+(1-W_i)\left(\tilde{y_i}-\hat{\mu}_2\right)\right\} \\ (1-\tilde{U}_i)\left\{W_i\left[\frac{-1}{\hat{\sigma}_2}+\frac{1}{\hat{\sigma}_2^3}(z_i-\hat{\mu}_2)^2\right]+(1-W_i)\left[\frac{-1}{\hat{\sigma}_2}+\frac{1}{\hat{\sigma}_2^3}D_4\right]\right\} \end{bmatrix} \tag{3.62}$$

Now, using (3.13), (3.50), (3.51) and (3.62), one can find I_L. Standard errors of the parameter estimates are the non-negative square root of the diagonal elements of I_L^{-1}.

3.7 Simulation Study

The main goal of this simulation study is to explore the benefits/downsides of study designs employing progressive Type-II censored mixture data and we consider two aspects. The first one focuses on the impacts of censoring in heterogeneous populations, and the second, makes an extensive comparison over four different scenarios which are discussed later in Section 3.7.4.

3.7.1 First simulation study

To illustrate the methods presented in Section 3.6 and show the advantages/disadvantages of the progressive Type-II censoring scheme, we carry out a simulation study. Normal mixture data under Type-II and progressive Type-II censoring are generated and the maximum likelihood estimates and their associated standard errors are found. We provide a number of tables containing the results of different simulation scenarios.We have developed the `R` package `pcensmix` which is used for the simulation study. The simulation study is structured as follows:

- Data from a two component normal mixture is generated under Type-II and progressive Type-II censoring under different true parameter values. Here we just report results for a dataset of size $N = 160$.

- Means and standard deviations for the distributions to be simulated from are chosen in such a way as to be able to explore the efficiency of the estimates and the identifiability of the components in well- and less- separated mixture populations and in the presence or absence of censoring. Typically, when the component means are far from each other, identifying the components is easier, however, when data are subject to Type-II or progressive Type-II censoring, this could be problematic and the choice of component standard deviations becomes more important to provide sufficient information from the components for estimation purposes. This will be discussed in more detail later.

- We consider six different sets of true parameter values for simulation from the two normal components as follows:

  (i) $\mu_1 = 10$, $\sigma_1 = 2$, $\mu_2 = 19$, $\sigma_2 = 4$, $\pi = 0.3$

  (ii) $\mu_1 = 10$, $\sigma_1 = 2$, $\mu_2 = 17$, $\sigma_2 = 4$, $\pi = 0.3$

  (iii) $\mu_1 = 10$, $\sigma_1 = 2$, $\mu_2 = 14$, $\sigma_2 = 4$, $\pi = 0.3$

  (iv) $\mu_1 = 10$, $\sigma_1 = 4$, $\mu_2 = 14$, $\sigma_2 = 4$, $\pi = 0.3$

  (v) $\mu_1 = 10$, $\sigma_1 = 2$, $\mu_2 = 25$, $\sigma_2 = 4$, $\pi = 0.3$

  (vi) $\mu_1 = 20$, $\sigma_1 = 6$, $\mu_2 = 22$, $\sigma_2 = 12$, $\pi = 0.6$

  These mixture densities are illustrated in Figure 3.1. We see that in the first two graphs at the top, components are quite identifiable especially when the difference between the mean components is larger. The other two graphs in the middle row, show that when the mean components are in the same distance, having a smaller standard deviation, at least for one component, makes the identifiability of mixture components easier and so we expect to see a better fitted model. The last two graphs at the bottom imply that having the mean components separate enough, if the components' standard deviations are so large, it would be problematic for model fitting as makes the component's densities overlap, whereas choosing smaller standard deviations separate out the components and give a bi-modal mixture density.

- The number of observed failures, r, is chosen to be equal to 160, 140, 130, 120 or 80, giving censoring fractions from 0 to 50%. Standard Type-II censoring scheme simply requires all observations greater than the r-th observed order statistic to be censored at the r-th largest sample failure time.

- The algorithm for generating progressive Type-II censored datasets is as follows:

  Algorithm.

  ① Compute the order statistics $(z_{(1)}, \ldots, z_{(N)})$ of the two component normal mixture sample $z_1, \ldots, z_N$ as the *Active Set* of times. They are all subject to either failure or censoring.

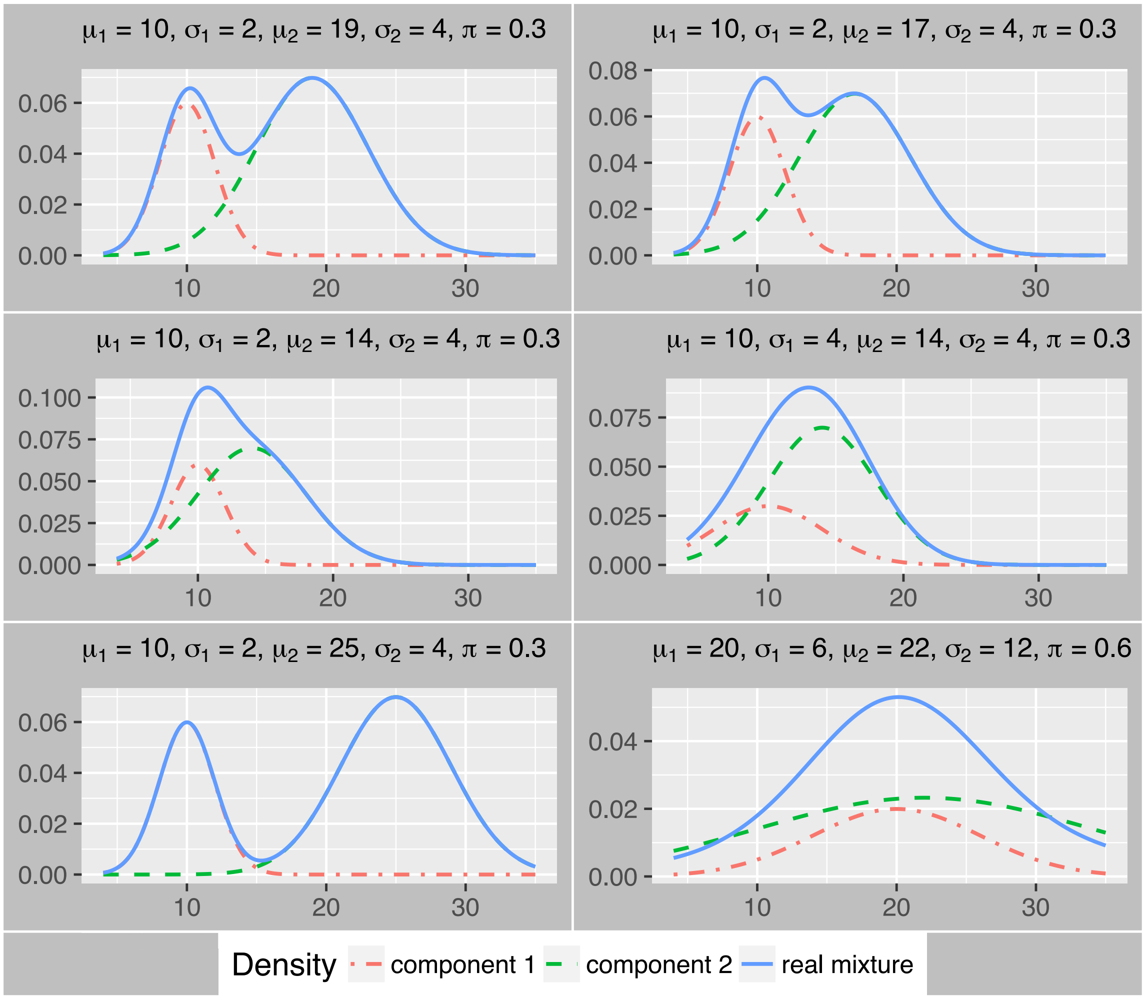


Figure 3.1: Illustration of mixture densities with different true values.

② Let $\boldsymbol{W} = (W_1, \ldots, W_N)$ be the censoring indicator.

③ Let $\boldsymbol{z}^{cv}$ represent the censored version of data, i.e., z_i^{cv} is equal to the actual observed survival time in the event of failure and is equal to the latest observed failure time if it is associated with an unobserved censored observation. In other words, here we treat the censored observations as observed since censored observations are still informative but not as much as complete failures.

④ Generate a random sampling plan $\mathcal{R} = (R_1, \ldots, R_r)$ such that $R_i \sim Bin(N - i - \sum_{j=1}^{r-1} R_j, p), j = 1, \ldots, r-1$, and $R_r = N - r - \sum_{j=1}^{r-1} R_j$.

⑤ Initialize the failure process setting, $i = 1$.

⑥ Form the *Observed Set* consisting of failures at stage i.

⑦ Let $\mathcal{S}$ be the *Risk Set*, a set of remaining survival times in the *Active Set* after removing the current *Observed Set* from it.

⑧ Generate without replacement, a random sample $\mathcal{Q}$ of size R_i from $\mathcal{S}$.

⑨ Update $\mathcal{S}$ by removing the generated sample $\mathcal{Q}$ from $\mathcal{S}$, i.e., $\mathcal{S} = \mathcal{S} \setminus \mathcal{Q}$.

⑩ Update $\boldsymbol{z}^{cv}$, i.e., first, combine $\boldsymbol{z}^{cv}$ with the i-th failure time appended on the right, then let all the censored order statistics $\mathcal{Q}$ be equal to the time of the i-th failure and merge them with the current elements of $\boldsymbol{z}^{cv}$ from its right-hand side.

⑪ If $i \leq r-1$, update $\boldsymbol{W}$, i.e., label the newly included elements of $\boldsymbol{z}^{cv}$ by 1 if not censored and 0 otherwise and store them in $\boldsymbol{W}$ in order; Then set $i = i+1$ and if $i = r$ go to ⑫, else repeat the steps from ⑥ to ⑪.

⑫ Update $\boldsymbol{z}^{cv}$ and $\boldsymbol{W}$ after observing the r-th failure, i.e., put all observations in $\mathcal{S}$, to be censored at the r-th failure, equal to the time of r-th failure and append them all, including the r-th failure, to $\boldsymbol{z}^{cv}$. Furthermore, include their censoring indicator values in $\boldsymbol{W}$ in the relevant order.

We use the EM algorithm introduced in Section 3.6.1 to fit the model and obtain matrix I_L (3.13) to compute the standard errors.

In fitting the model, to overcome the problem of singularity and model non-identifiability that might happen in some cases, depending on the true means and standard deviations of the components, we use the constraint proposed by Hathaway (1985). Based on this, the ratios of the standard deviations are considered not to be less than a pre-fixed constant. Here we consider $\frac{\sigma_1}{\sigma_2} > 0.1$ and $\frac{\sigma_2}{\sigma_1} > 0.1$ which provide a constrained parameter space with no singularities. Note that in the simulation study, at the iteration ν of the EM algorithm, we re-parametrize the weights of component membership estimations by $(1-\epsilon)\tilde{U}_i + \epsilon$ to try and reallocate the observations to the mixture components and prevent the majority of them belonging to one component; we use $\epsilon = 0.95$. Furthermore, if the above constraints are violated at the iteration ν, we generate a new dataset to avoid the algorithm getting into the singular likelihood area and at the same time, rescale ϵ by $\log_\nu \epsilon$ so that the value of ϵ increases at a slow rate in next iterations where the conditions may be again violated.

Following model fitting with the EM algorithm, the adequacy of the model should be evaluated in terms of the feasibility of parameter estimates and the deviance criterion, $-2\times$log-likelihood, and also the availability of the standard errors. As described before, we have considered different true parameter sets for the simulation study to investigate the effect of the mean and variance of the mixture components and also the effect of Type-II and progressive Type-II censoring schemes on parameter estimates and their standard errors, to observe the situations that increase the model efficacy and to illustrate the benefits of progressive censoring regime. In Table 3.1, parameter estimates as well as their associated standard errors are shown for three different true parameter sets when the number of failures, r, is 130. For each set the test-time duration for the Type-II censoring plan is the smallest, however, the efficiency of estimates under progressive Type-II censoring is higher in terms of having smaller standard errors. [Table 3.1 goes here]

Looking at the standard errors obtained under the progressive regime, we can see that as the degree of progressive censoring increases, as controlled by the parameter p, the test time duration increases but the standard errors typically become smaller. This clearly shows that when keeping the pre-specified number of failures at the same number, removing experimental individuals at random with a progressive sampling scheme increases the estimation efficiency by spreading out the censored observations over both components and so making a better balance between the amount of censoring from each component. This feature of the progressive Type-II censoring scheme is very beneficial in designing a cost-effective experiment by choosing a small number of failed experimental individuals to observe and considering a higher degree of censoring and at the same time achieving higher relative efficiency for the estimates. This is shown in Table 3.2 when considering small values for the number of failures, r. [Table 3.2 goes here]. It is noteworthy to mention that the above finding can be seen in Table 3.4 as well.

Depending on the choice of true parameters, when fitting a model under Type-II censoring with repeated simulations, we observed that there is an approximately 1% chance of failing to obtain a standard error for at least one of the parameter estimators. This happens especially when the mean components are not well-separated enough. It is worth pointing out that censored observations are not non-informative but partially informative and hence useful to some extent. Therefore, apart from the effect of observed complete-failures in making the parameter estimation feasible, censored observations from both components are also of great importance especially in the mixture structure, otherwise, the feasibility of obtaining the estimates would be lost due to lack of information. Hence, the reason of this failure in obtaining the standard errors in these cases, is having most, or almost all, censored observations from the component with the larger mean life-time. However, progressive Type-II censoring improves the situation by spreading out the censoring over the two components. For instance, for the first parameter set mentioned in Table 3.1, when $p = 0$ (i.e. Type-II censoring), there is 0.759 censored experimental individuals from the first component, which is very small number, and 29.240 units censored from component two. For the same case, but under progressive Type-II censoring when $p = 0.3$, the number of censored observations from the first and the second components are 6.410 and 23.590 respectively. Comparing these two situations shows the main benefit of a progressive censoring regime with the ability of better handling the mixture model fitting and identifiability by protecting against this problem. This is consistent with Policello II (1981) where he fits a constrained EM algorithm by conditioning on there being at least two observations from each mixture component to avoid singularities.

Table 3.3 shows the parameter and standard error estimates when the number of observed failures is considered equal to 140. The results show the same pattern as in Table 3.1, which is a reconfirmation of the higher capability of the progressive Type-II censoring scheme comparing to Type-II censoring scheme in handling the mixture model fitting. [Table 3.3 goes here]

In terms of the speed of computation, as p declines, the convergence is typically slower. In other words, computational speed is higher when the censored observations are spread out over the components more evenly and model fitting under Type-II censoring, $p = 0$, is computationally the slowest case for any particular set of simulation.

Table 3.4 shows that the parameter estimates are very close to the real parameters values with small standard errors, when the components are very well-separated with relatively small standard deviations. Note that in this case, the algorithm identifies the components very quickly in each iteration, and hence the computational speed is really high. [Table 3.4 goes here]

3.7.2 Tables of results

Results obtained from the first simulation study are summarized in the tables provided in this Section.

Table 3.1: Two component normal mixture: parameter estimates with $N = 160$ and $r = 130$, using mean values.

True values	p	Test-time	-2 loglik	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
	0†	16.478	969.890	10.110	1.908	14.800	3.537	0.394	62.275	67.725	0.759	29.241
μ_1=10				(0.915)	(0.607)	(0.704)	(1.043)	(0.248)				
σ_1=2	0.20	19.386	958.004	10.326	1.902	15.122	3.445	0.429	62.966	67.033	5.698	24.302
μ_2=14				(0.628)	(0.459)	(1.166)	(0.657)	(0.155)				
σ_2=4	0.30	21.548	969.284	10.169	1.998	15.023	3.518	0.425	61.657	68.343	6.410	23.590
π=0.3				(0.775)	(0.553)	(1.385)	(0.637)	(0.199)				
	0††	19.436	990.236	10.338	2.069	17.542	3.501	0.374	59.702	70.298	0.116	29.884
μ_1=10				(0.752)	(0.474)	(1.065)	(0.849)	(0.132)				
σ_1=2	0.20	22.571	993.836	10.285	2.041	17.410	3.704	0.349	52.886	77.114	2.917	27.082
μ_2=17				(0.737)	(0.484)	(1.031)	(0.654)	(0.124)				
σ_2=4	0.30	24.344	985.956	10.233	2.002	17.483	3.621	0.354	53.336	76.664	3.312	26.688
π=0.3				(0.693)	(0.445)	(0.958)	(0.608)	(0.113)				
	0	27.586	1238.186	18.593	5.468	19.977	10.690	0.621	86.087	43.913	13.273	16.727
μ_1=20				(1.48)	(1.352)	(6.926)	(4.277)	(0.253)				
σ_1=6	0.20‡	36.678	1253.472	19.814	5.518	21.418	11.146	0.592	80.525	49.475	14.233	15.766
μ_2=22				(1.272)	(1.264)	(5.639)	(3.636)	(0.222)				
σ_2=12	0.30	41.279	1249.604	19.943	5.522	22.526	10.200	0.609	81.720	48.280	15.680	14.320
π=0.6				(1.210)	(1.199)	(3.756)	(2.582)	(0.208)				

For each box, the first row presents the estimates for the corresponding parameters and the second row is their standard error within parentheses.

†Results in this part are based on 99 repetitions due to nonavailability of SE($\hat{\mu_2}$) and SE($\hat{\sigma_2}$) in one repetition.

††Based on 95 repetitions.

‡ Based on 96 repetitions.

Table 3.2: Two component normal mixture: parameter estimates with $N = 160$, using mean values.

	True values	p	Test-time	-2 loglik	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
$r = 120$	$\mu_1 = 10$	0.20†	17.339	948.492	10.185	1.999	15.135	3.237	0.435	63.277	56.722	6.374	33.626
	$\sigma_1 = 2$				(0.933)	(0.570)	(1.563)	(0.947)	(0.230)				
	$\mu_2 = 14$	0.30	30.262	1000.208	10.007	1.829	15.025	3.502	0.398	45.495	74.505	2.272	37.728
	σ_2=4				(0.676)	(0.486)	(1.240)	(0.746)	(0.163)				
	π=0.3												
$r = 120$	$\mu_1 = 20$	0.20††	29.461	1248.714	18.074	5.415	20.011	9.833	0.580	74.186	45.814	18.692	21.308
	$\sigma_1 = 6$				(1.571)	(1.437)	(5.253)	(3.291)	(0.275)				
	$\mu_2 = 22$	0.3‡	34.663	1242.54	19.694	5.306	19.471	10.103	0.567	71.343	48.657	19.404	20.596
	σ_2=12				(3.027)	(1.741)	(5.500)	(3.370)	(0.454)				
	π=0.6												
$r = 100$	$\mu_1 = 10$	0.20‖	14.935	967.46	8.468	2.919	13.806	3.175	0.313	38.578	61.422	11.569	48.431
	$\sigma_1 = 4$				(2.977)	(1.344)	(1.714)	(1.204)	(0.401)				
	$\mu_2 = 14$	0.3‡‡	15.476	971.096	8.335	3.155	13.767	3.388	0.284	34.264	65.736	11.195	48.805
	σ_2=4				(2.912)	(1.522)	(1.770)	(1.125)	(0.343)				
	π=0.3												

For each box, the first row presents the estimates for the corresponding parameters and the second row is their standard error within parentheses.

For the case of not very well separated component means, when the number of failures is small, as the amount of censoring gets larger the efficiency in estimation gets higher and the EM algorithm is computationally faster to converge.

†Results based on 99 repetitions due to nonavailability of SE($\hat{\mu}_2$) and SE($\hat{\sigma}_2$) in one repetition.

††Based on 96 repetitions.

‡Based on 97 repetitions.

‖ Results based on 93 repetitions.

‡‡ Results based on 95 repetitions.

Table 3.3: Two component normal mixture: parameter estimates with $N = 160$ and $r = 140$, using mean values.

True values	p	Test-time	-2 loglik	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
	0†	17.723	967.862	10.135	1.967	15.153	3.341	0.432	68.913	71.087	0.296	19.704
μ_1=10				(0.733)	(0.475)	(1.346)	(0.836)	(0.180)				
σ_1=2	0.20	22.038	972.274	10.208	1.977	15.096	3.491	0.415	62.360	77.639	4.130	15.870
μ_2=14				(0.751)	(0.503)	(1.298)	(0.598)	(0.175)				
σ_2=4	0.3	22.491	970.45	10.092	1.885	15.060	3.516	0.406	60.861	79.139	4.116	15.884
π=0.3				(0.617)	(0.505)	(1.226)	(0.539)	(0.168)				
	0	20.691	1001.128	10.278	2.034	17.492	3.643	0.361	82.324	0.069	19.931	57.676
μ_1=10				(0.694)	(0.446)	(1.002)	(0.710)	(0.120)				
σ_1=2	0.20	22.571	993.834	10.285	2.041	17.410	3.704	0.349	52.886	77.114	2.917	27.083
μ_2=17				(0.737)	(0.484)	(1.031)	(0.654)	(0.124)				
σ_2=4	0.30	25.833	996.08	10.176	1.918	17.381	3.770	0.337	52.012	87.988	1.972	18.028
π=0.3				(0.605)	(0.416)	(0.902)	(0.554)	(0.103)				
	0††	30.201	1243.276	19.205	5.271	20.060	9.824	0.564	83.667	56.333	6.628	13.371
μ_1=20				(1.528)	(1.331)	(19.124)	(5.890)	(0.247)				
σ_1=6	0.20‡	42.914	1256.304	20.289	5.759	21.068	10.514	0.618	88.155	51.845	10.662	9.337
μ_2=22				(1.167)	(1.160)	(3.915)	(2.758)	(0.192)				
σ_2=12	0.30	45.756	1264.684	20.128	5.462	22.676	11.505	0.599	85.713	54.286	10.080	9.920
π=0.6				(1.200)	(1.246)	(4.151)	(2.864)	(0.231)				

For each box, the first row presents the estimates for the corresponding parameters and the second row is their standard error within parentheses.

†Results in this part are based on 96 repetitions due to nonavailability of SE($\hat{\sigma}_2$) in four repetition.

††Results based on 98 repetitions.

‡ Results based on 99 repetition.

Table 3.4: Parameter estimates with $N = 160$, using mean values.

	True values	p	Test-time	-2 loglik	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
$r = 80$	$\mu_1 = 10$	0.20	23.664	998.108	9.703	1.593	25.174	4.633	0.261	40.304	39.696	1.416	78.584
	$\sigma_1 = 2$				(0.523)	(0.475)	(1.772)	(3.704)	(0.053)				
	$\mu_2 = 25$	0.30	23.871	1922.418	9.934	2.043	24.604	3.112	0.266	40.834	39.166	1.837	78.163
	σ_2=4				(0.324)	(0.230)	(0.517)	(0.603)	(0.036)				
	π=0.3												
$r = 120$	$\mu_1 = 10$	0.20	28.302	1005.520	10.037	1.974	24.948	4.035	0.294	45.475	74.525	1.535	38.464
	$\sigma_1 = 2$				(0.322)	(0.245)	(0.436)	(0.411)	(0.038)				
	$\mu_2 = 25$	0.30	30.262	1000.208	9.973	1.978	24.902	3.999	0.298	45.495	74.505	2.272	37.728
	σ_2=4				(0.323)	(0.245)	(0.442)	(0.394)	(0.038)				
	π=0.3												
$r = 130$	$\mu_1 = 10$	0.20	30.587	1007.000	9.955	1.978	24.967	3.993	0.297	45.969	84.030	1.502	28.498
	$\sigma_1 = 2$				(0.314)	(0.237)	(0.421)	(0.360)	(0.037)				
	$\mu_2 = 25$	0.30	32.344	1003.972	9.960	1.959	24.992	3.984	0.299	46.055	83.945	1.728	28.272
	σ_2=4				(0.313)	(0.238)	(0.426)	(0.356)	(0.038)				
	π=0.3												

For each box, the first row presents the estimates for the corresponding parameters and the second row is their standard error within parentheses.

For the case of very well separated component means, the estimates are more efficient and fitting models is computationally faster.

3.7.3 Discussion

Although the EM algorithm is known to be a useful practical method for fitting a model to mixture data, it requires care and effort, especially when having normal mixture data. This is due to the problem of singularity of the likelihood function at the parameter space boundary when maximizing the likelihood. This problem is rather serious when the data are influenced by some sort of censoring, as in reliability or medical experiments. Under Type-II censoring, the follow-up is terminated after a pre-specified number of failures, with all other individuals censored at the largest failure time. In this case, the likelihood function is computationally expensive to maximize and often does not lead to very efficient and reliable parameter and standard error estimates. It is sensitive to the choice of the initial starting mixing proportion parameter as with a small value we could end up having no complete failure observations from a mixture component and then as a result getting poor, or sometimes no, sensible estimates for parameters from that component. In progressive Type-II censoring some individuals are removed randomly at each failure time. This has the effect of spreading out the censoring over the observation period with a consequent extension of the follow-up time. However, this censoring scheme improves the efficiency of estimation for mixed populations and leads to more sensible estimates for both components. Moreover, the problem of singularity and model non-identifiability might happen in some cases depending on the true means and standard deviations of the components. The constrained EM algorithm proposed by Hathaway (1985) is a useful method to use in this case to obtain a parameter space with no singularities.

In this chapter, we have considered modeling a single heterogeneous population when the data arises from a Type-II or progressive Type-II censoring scheme and discussed the possible benefits/downsides of a progressive censoring regime. This can be extended to two heterogeneous populations applying Type-II or progressive Type-II censoring over the two mixture populations, referred to as joint (progressive) Type-II censoring schemes. Generally, under joint progressive censoring, the sample is based on two baseline samples $X_1, \ldots, X_{n_1}$ (product/Type A) and $Y_1, \ldots, Y_{n_2}$ (product/Type B) of independent random variables, where the X-sample follows F_1 distribution and Y-sample follows F_2 distribution and the progressive censoring is applied over the pooled sample, with the assumption that the type of failures and the type of censored individuals are known. Rasouli and Balakrishnan (2010) considered such homogeneous joint populations for inference under joint progressive Type-II censoring when F_1 and F_2 are both exponential. However, it would be possible to extend this to include heterogeneous (mixed) populations for the samples.

One of the benefits of mixture models is that when mixing probabilities are directly related to some explanatory variables, we can include them in the model as covariates associated with the survival times. Kuo and Peng (2000) use a logistic regression link for the mixing proportions π for modeling male beetle data where beetles are exposed to four particular dosages of toxicity. This can also be used in the joint censoring set-up for modeling each mixed population.

Other than using normal distributions, we could assume mixture data from other lifetime distributions such as exponential, Weibull, log-normal or gamma. More generally, fitting a mixture of generalized linear models (GLMs) could be considered. In such cases, using a classic EM algorithm might be hard in terms of evaluating the E-step of the first EM algorithm where we need to integrate the complete-data log-likelihood over the censored observations given the observed data rather than just replacing the conditional expected value of the mixture components in the complete-data log-likelihood. The Monte-Carlo EM algorithm (MCEM) can then be used to calculate the E-step numerically with Monte Carlo simulation as suggested in Kuo and Peng (2000).

In terms of application of such progressive Type-II schemes, there are obvious benefits in machine testing in industrial applications where effort, resource, and cost can be saved by early censoring. Indeed although such schemes lead to longer average test times, the associated cost could be much reduced and costs can be used in planning the censoring plan $\mathcal{R}$. Similarly, in the social sciences, we could consider using such schemes in studies of recidivism, where continuing follow-up may be costly. One could also envisage the use of such a scheme in cohort studies of particular patient groups, where again long-term monitoring may be costly. The joint version of the scheme is ideal for comparative experiments of say different types of machines, different follow-up care policies on release from prison, and clinical trials involving longer term follow-up.

3.7.4 Second simulation study

In this section we carry out a simulation study to illustrate the methods represented in Sections 3.3 to 3.6. The simulation study at this stage aims to compare the results from four different scenarios as follow:

- We develop a function in `R` for each model fitting – As described before the six comparative fittings are done when

    1 labels of the population individuals are assumed to be known. This is possible since the data are simulated. In this case, the data are complete with no censoring.

2 labels are known but some observations are Type-II censored or progressively Type-II censored. In this case, censored data with known labels provide the information about the virtual mixture component that the relevant censored individual is from. Having that together with the censoring time information (e.g. the last observed failure time for Type-II censoring), we can estimate the censored life-times and squared censored times using formulas (i) and (ii) in the Proposition in Section 3.2. Then we find the maximum likelihood parameter estimates with EM algorithm by considering the observed likelihood function.

3 we have a population of mixture of normals.

4 we have a mixture Type-II censored normal and a mixture progressively Type-II censored normal data. These are similar to both models explained in number 2 above, except we consider a component indicator as a latent variable which needs to be estimated in the first EM algorithm and moreover censored life-times and also squared life-times are estimated in the second EM algorithm. In this case due to the presence of latent variables, we consider the complete-data likelihood for the maximum likelihood estimation purpose.

- For all the above mentioned scenarios, we have also done direct optimization using `optim` function in `R`. This function performs minimization by default, so we use the negative of the observed log-likelihood function in order to get `optim` to do the maximization. The other way of doing this is scaling the target log-likelihood function using `fnscale` as a control argument in `optim` - This function will maximize if `fnscale` is negative.

  The `optim` function in `R` provides some options for the optimization methods to be used. To make sure the estimates of σ_1 and σ_2 are range preservative, we use the `L-BFGS-B` method and set a constraint in this regard. This can avoid happening the possible non-invertibility problem of the Hessian matrix. Notice that the default method of optimization is `Nelder-Mead`. For more details, see help manual of `optim` in package (R Core Team, 2014, stats)

- Data can be generated under Type-II censoring and progressive Type-II censoring in a similar way as in the algorithm described in the first simulation study, with different true parameter values.

- We consider different sets of true parameter values for data simulation as follows:

  (i) $\mu_1 = 10$, $\sigma_1 = 2$, $\mu_2 = 19$, $\sigma_2 = 4$, $\pi = 0.3$

  (ii) $\mu_1 = 10$, $\sigma_1 = 2$, $\mu_2 = 17$, $\sigma_2 = 4$, $\pi = 0.3$

and the number of observed failures, r, are chosen to be equal to 160, 140, 120 and 80. Notice that the choice of $r = 160$, as in Table 3.5, corresponds to no censoring case and is considered in the simulation study to show that the results obtained from fitting a mixture model and a theoretical censored mixture model (although in practice no censoring has occurred) are the same in this case.

- Different values are considered for the number of observed failures. Obviously the case of having the amount of failure as high as the population size, shows the results of non-censored data.

- As mentioned before, one of the ways we use for the estimation of parameters and their standard errors is direct optimization technique. In some extreme cases, the Hessian matrix returned by the `optim` function is not invertible duo to the feature of the generated Type-II censored data. It includes the case that in the Type-II censored data we happen to have either no failure from one mixture component and all from the other component or all censored observations from one component. (remember to write about singularity and positive-definiteness)

- The samples that lead to singularity in the likelihood function and hence the problem of non-invertibility of the Hessian matrix returned by `optim` function and the information matrix returned by the Louis method are discarded. This is not likely to happen for all the fitted models to this data at the same time. Hence, we use the function `tryCatch` in `R` which provides a mechanism for handling unusual conditions, including errors and warnings. Doing so, we generate a new data and fit all the models to the new data.

- The chosen value of the mixing proportion π in the simulation study plays an important role in making a mixture model identifiable or non-identifiable. This problem is marked in the presence of censoring. That is, the higher value of π indicates the higher proportion of failures from the first component.

3.7.5 Table of results

Results obtained from the second simulation study are summarized in the tables provided in this Section.

Table 3.5: Parameter estimates with $N = 160$, $r = 160$, using mean values

			Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	19	4	0.3	48	112	0	0
		Sample(+lab)	29.432	9.979	1.976	19.075	4.030	0.302	48.360	111.640	0	0
		Sample(lab+TIIcens)										
Method	LL											
Optim	-414.074		29.431	9.979	1.955	19.07	4.012		48.360	111.640	0	0
				(0.282)	(0.200)	(0.380)	(0.269)					
EM	-414.073		29.431	9.979	1.955	19.07	4.012					
				(0.282)	(0.200)	(0.380)	(0.269)					
		Sample(lab+PTIIcens)										
Method	LL											
Optim	-413.790		29.374	9.989	1.945	19.068	3.995		48.000	112.000	0	0
				(0.279)	(0.197)	(0.376)	(0.266)					
EM	-413.790		29.374	9.989	1.945	19.068	3.995					
				(0.279)	(0.197)	(0.376)	(0.266)					
		Mixt(no lab)										
Method	LL											
Optim	-485.757		29.431	10.028	1.950	19.085	3.990	0.303	48.360	111.640	0	0
				(0.489)	(0.349)	(0.658)	(0.452)	(0.065)				
EM	-510.535		29.431	10.082	2.000	19.092	3.917	0.313				
				(0.489)	(0.349)	(0.658)	(0.452)	(0.065)				
		Mixt(+cens)										
Method	LL											
Optim	-485.757		29.432	10.027	1.950	19.085	3.990	0.303	48.571	111.428	0	0
				(0.489)	(0.349)	(0.658)	(0.452)	(0.065)				
EM	-510.512		29.432	10.027	1.950	19.085	3.990	0.303				
				(0.489)	(0.349)	(0.658)	(0.452)	(0.065)				
		Mixt(+Pcens)										
Method	LL											
Optim	-485.757		29.431	10.027	1.950	19.085	3.990	0.303	48.571	111.428	0	0
				(0.490)	(0.349)	(0.658)	(0.451)	(0.065)				
EM	-510.512		29.431	10.027	1.950	19.085	3.990	0.303				
				(0.490)	(0.349)	(0.658)	(0.451)	(0.065)				

Using Mean Values

Table 3.6: Parameter estimates with $N = 160$, $r = 140$ and $R = c(rep(0, (r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

Method	LL		Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	19	4	0.3	48	112	0	0
		Sample(+lab)	29.098	9.943	2.007	18.959	3.958	0.309	49.380	110.620	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-376.739		22.496	9.943	1.986	18.955	3.933		49.380	90.620	0.000	20.000
				(0.284)	(0.201)	(0.384)	(0.306)					
EM	-409.471		22.496	9.943	1.986	18.955	3.933					
				(0.284)	(0.201)	(0.384)	(0.306)					
		Mixt(no lab)										
Method	LL											
Optim	-483.892		29.098	9.986	1.963	19.033	3.882	0.315	49.380	110.620	0	0
				(0.491)	(0.339)	(0.656)	(0.451)	(0.065)				
EM	-507.844		29.098	9.986	1.963	19.033	3.882	0.315				
				(0.491)	(0.339)	(0.656)	(0.451)	(0.065)				
		Mixt(+cens)										
Method	LL											
Optim	-448.694		22.496	10.035	1.987	19.058	3.812	0.320	51.183	88.816	0.001	19.999
				(0.515)	(0.352)	(0.667)	(0.578)	(0.070)				
EM	-504.166		22.496	10.035	1.987	19.058	3.812	0.320				
				(0.515)	(0.352)	(0.667)	(0.578)	(0.070)				

Using Mean Values

Table 3.7: Parameter estimates with $N = 160$, $r = 120$ and $R = c(rep(0,(r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

			Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	19	4	0.3	48	112	0	0
		Sample(+lab)	29.180	10.010	2.000	18.927	3.994	0.298	47.700	112.300	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-334.994		20.326	10.010	1.979	18.912	3.956		47.700	72.300	0.000	40.000
				(0.288)	(0.204)	(0.411)	(0.357)					
EM	-407.337		20.326	10.010	1.979	18.912	3.956					
				(0.284)	(0.201)	(0.384)	(0.306)					
		Mixt(no lab)										
Method	LL											
Optim	-483.646		29.180	10.042	1.935	18.964	3.928	0.300	47.700	112.300	0	0
				(0.512)	(0.373)	(0.674)	(0.457)	(0.068)				
EM	-508.361		29.180	10.042	1.935	18.964	3.928	0.300				
				(0.512)	(0.373)	(0.674)	(0.457)	(0.068)				
		Mixt(+cens)										
Method	LL											
Optim	-404.682		20.326	10.134	1.965	19.015	3.729	0.313	50.260	69.739	0.002	39.997
				(0.577)	(0.392)	(0.676)	(0.771)	(0.078)				
EM	-497.261		20.326	10.134	1.965	19.015	3.729	0.313				
				(0.578)	(0.398)	(0.681)	(0.775)	(0.079)				

Using Mean Values

Table 3.8: Parameter estimates with $N = 160$, $r = 80$ and $R = c(rep(0,(r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

Method	LL		Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	19	4	0.3	48	112	0	0
		Sample(+lab)	28.999	10.016	1.972	18.982	3.961	0.302	48.410	111.590	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-224.939		16.586	10.016	1.952	18.800	3.805		48.340	31.660	0.070	79.930
				(0.282)	(0.200)	(0.667)	(0.564)					
EM	-386.568		16.586	10.016	1.952	18.800	3.805					
				(0.282)	(0.200)	(0.667)	(0.564)					
		Mixt(no lab)										
Method	LL											
Optim	-483.548		28.999	10.106	1.961	19.095	3.862	0.313	48.410	111.590	0	0
				(0.501)	(0.352)	(0.648)	(0.446)	(0.065)				
EM	-507.655		28.999	10.106	1.961	19.095	3.862	0.313				
				(0.501)	(0.352)	(0.648)	(0.446)	(0.065)				
		Mixt(+cens)										
Method	LL											
Optim	-296.516		16.633	10.1742	1.965	18.529	3.138	0.324	50.716	29.284	0.227	79.773
				(0.576)	(0.387)	(0.855)	(1.371)	(0.085)				
EM	-460.353		16.633	10.163	1.942	18.554	3.264	0.318				
				(0.581)	(0.394)	(0.891)	(1.397)	(0.086)				

Using Mean Values

Table 3.9: Parameter estimates with $N = 160$, $r = 160$ and $R = c(rep(0, (r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

Method	LL		Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	17	4	0.3	48	112	0	0
		Sample(+lab)	27.477	9.964	1.988	16.972	4.027	0.300	48.120	111.880	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-414.361		27.477	9.964	1.967	16.972	4.010		48.120	111.880	0.000	0.000
				(0.285)	(0.202)	(0.379)	(0.268)					
EM	-414.361		27.477	9.964	1.967	16.972	4.010					
				(0.285)	(0.202)	(0.379)	(0.268)					
		Mixt(no lab)										
Method	LL											
Optim	-468.463		27.477	10.173	2.013	17.331	3.806	0.342	48.120	111.880	0	0
				(0.688)	(0.441)	(1.021)	(0.549)	(0.119)				
EM	-506.936		27.477	10.173	2.013	17.331	3.806	0.342				
				(0.688)	(0.441)	(1.021)	(0.549)	(0.119)				
		Mixt(+cens)										
Method	LL											
Optim	-468.463		27.477	10.173	2.013	17.331	3.806	0.342	54.758	105.241	0.000	0.000
				(0.689)	(0.441)	(1.022)	(0.549)	(0.120)				
EM	-506.923		27.477	10.173	2.013	17.331	3.806	0.342				
				(0.689)	(0.441)	(1.022)	(0.549)	(0.120)				

Using Mean Values

Table 3.10: Parameter estimates with $N = 160$, $r = 140$ and $R = c(rep(0,(r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

			Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	17	4	0.3	48	112	0	0
		Sample(+lab)	27.098	9.943	2.007	16.959	3.958	0.309	49.380	110.620	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-376.739		20.496	9.943	1.986	16.955	3.933		49.380	90.620	0.000	20.000
				(0.284)	(0.201)	(0.384)	(0.307)					
EM	-409.472		20.496	9.943	1.986	16.955	3.933					
				(0.284)	(0.201)	(0.384)	(0.307)					
		Mixt(no lab)										
Method	LL											
Optim	-467.304		27.098	10.059	1.936	17.259	3.753	0.336	49.380	110.620	0	0
				(0.595)	(0.410)	(0.888)	(0.508)	(0.102)				
EM	-502.742		27.098	10.095	1.938	17.279	3.753	0.339				
				(0.588)	(0.408)	(0.888)	(0.507)	(0.101)				
		Mixt(+cens)										
Method	LL											
Optim	-432.035		20.496	10.181	1.963	17.384	3.580	0.357	57.458	82.541	0.035	19.964
				(0.629)	(0.421)	(0.913)	(0.672)	(0.110)				
EM	-496.176		20.496	10.184	1.979	17.406	3.561	0.359				
				(0.632)	(0.422)	(0.913)	(0.674)	(0.110)				

Using Mean Values

Table 3.11: Parameter estimates with $N = 160$, $r = 120$ and $R = c(rep(0,(r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

Method	LL		Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	17	4	0.3	48	112	0	0
		Sample(+lab)	27.180	10.01	2.000	16.927	3.994	0.298	47.700	112.300	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-335.000		18.327	10.010	1.978	16.912	3.956		47.690	72.310	0.010	39.990
				(0.288)	(0.204)	(0.411)	(0.357)					
EM	-407.329		18.327	10.010	1.978	16.912	3.956					
				(0.288)	(0.204)	(0.411)	(0.357)					
		Mixt(no lab)										
Method	LL											
Optim	-467.006		27.180	10.145	1.943	17.253	3.756	0.328	47.700	112.300	0	0
				(0.643)	(0.436)	(0.852)	(0.4816)	(0.101)				
EM	-503.185		27.180	10.183	1.966	17.308	3.739	0.334				
				(0.650)	(0.438)	(0.872)	(0.490)	(0.102)				
		Mixt(+cens)										
Method	LL											
Optim	-388.033		18.327	10.296	2.002	17.372	3.353	0.362	58.004	61.996	0.142	39.858
				(0.759)	(0.471)	(0.940)	(0.921)	(0.128)				
EM	-486.251		18.327	10.300	2.016	17.385	3.338	0.363				
				(0.764)	(0.473)	(0.942)	(0.927)	(0.128)				

Using Mean Values

Table 3.12: Parameter estimates with $N = 160$, $r = 80$ and $R = c(rep(0,(r-1)), m - r - \sum_{i=1}^{r-1} R_i))$

			Test-time	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{\pi}$	r_1	r_2	n_1	n_2
		True		10	2	17	4	0.3	48	112	0	0
		Sample(+lab)	26.810	10.008	1.977	17.034	3.972	0.299	47.800	112.200	0	0
		Sample(lab+cens)										
Method	LL											
Optim	-227.644		14.753	10.007	1.952	16.933	3.900		47.260	32.740	0.540	79.460
				(0.284)	(0.202)	(0.667)	(0.568)					
EM	-390.478		14.753	10.007	1.952	16.933	3.900					
				(0.284)	(0.202)	(0.667)	(0.568)					
		Mixt(no lab)										
Method	LL											
Optim	-467.421		26.810	10.268	1.994	17.482	3.680	0.345	47.800	112.200	0	0
				(0.603)	(0.412)	(0.849)	(0.490)	(0.100)				
EM	-502.777		26.810	10.268	1.994	17.482	3.680	0.345				
				(0.603)	(0.412)	(0.849)	(0.490)	(0.100)				
		Mixt(+cens)										
Method	LL											
Optim	-281.921		14.817	10.083	1.812	16.675	3.103	0.321	51.539	28.460	1.065	78.934
				(0.718)	(0.438)	(1.475)	(1.491)	(0.128)				
EM	-450.982		14.816	10.129	1.856	16.712	3.055	0.329				
				(0.802)	(0.465)	(1.553)	(1.632)	(0.141)				

3.8 Conclusions

In this chapter, we chose the Normal mixture distribution for estimation under (progressive) Type-II censoring which has not been studied before. According to the results obtained in Section 3.7.2 and 3.7.5, we come to this conclusion that one of the main benefits of progressive censoring as a smart sampling mechanism is that it lets us design experiments with less sacrifice of individuals and yet comes to more accurate results, though at the price of a longer experimental time than under standard Type-II censoring. We also observe that:

- Under Type-II censoring, we have poorer estimates for parameters from the component with bigger mean with large SE's.

- Progressive Type-II censoring seems to protect against heterogeneity and thus better estimates for parameters from both the components.

- Test time is longer under Progressive Type-II censoring, though shorter than no censoring case.

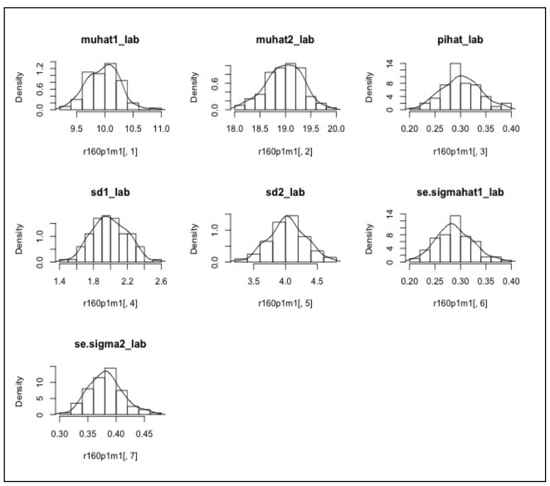
- As r, the number of complete failures that defines the Type-II censoring stopping rule, becomes larger, SE's become smaller, however, the experiment would take longer.

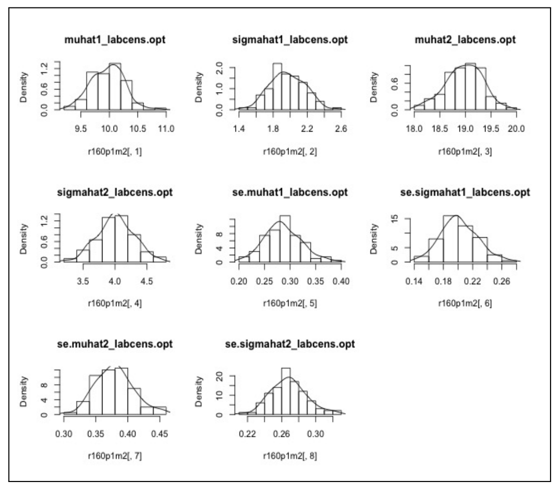
Notice that the results from the second simulation study are obtained in two ways: (i) direct optimization using the `R` function `optim` (ii) the developed EM algorithm. Equality of results from these methods to at least three decimals is a confirmation that the developed EM algorithm is reliable and accurate.

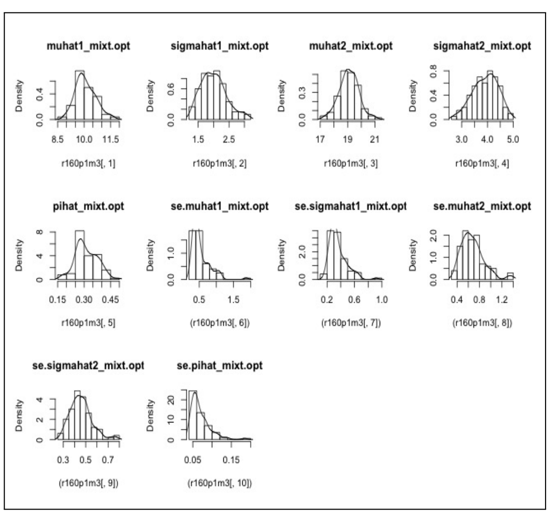
Histogram and density of estimates

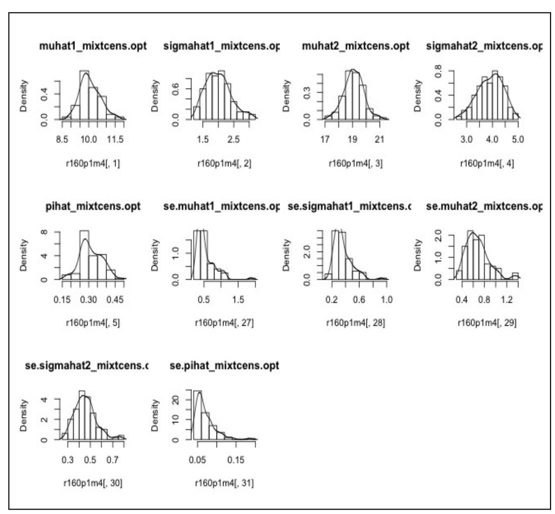
Here we see the histogram and density plots of the estimates across the models for different values of r and with different considered true values. For the sake of simplicity and saving space, we use the notation `<r · p · m ·>` for indicating the parameter estimates obtained from the specific values of r, true values and the underlying model. For example, `r160p1m1` indicates a matrix which its columns are the 100 obtained estimates with $r = 160$ and when using the first set of parameters as in (i) for fitting the first model. Obviously, using matrix notation in `R`, `r160p1m1[,i]` shows the i-th column of this matrix.

- when $r = 160$ and for parameter set i

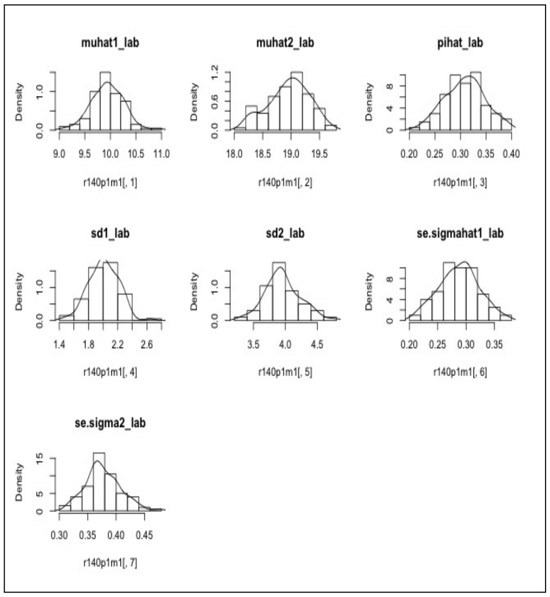


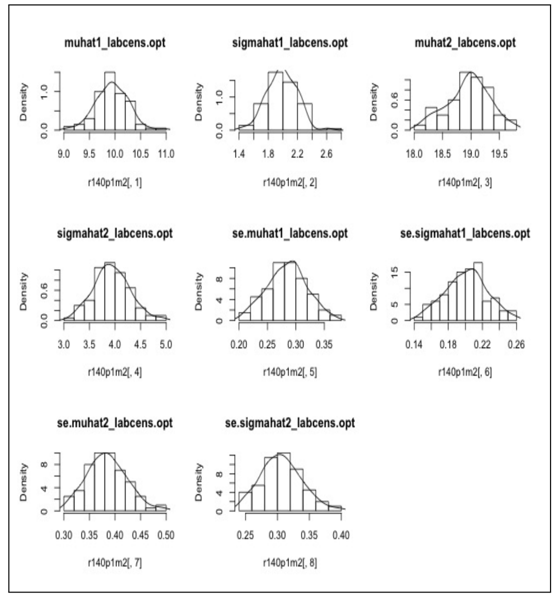


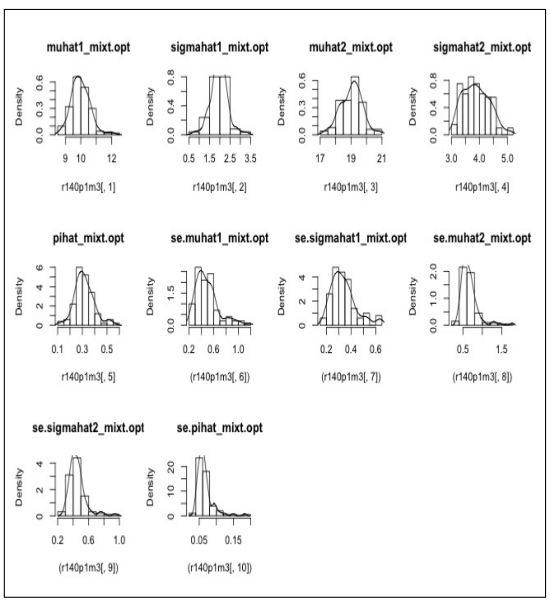


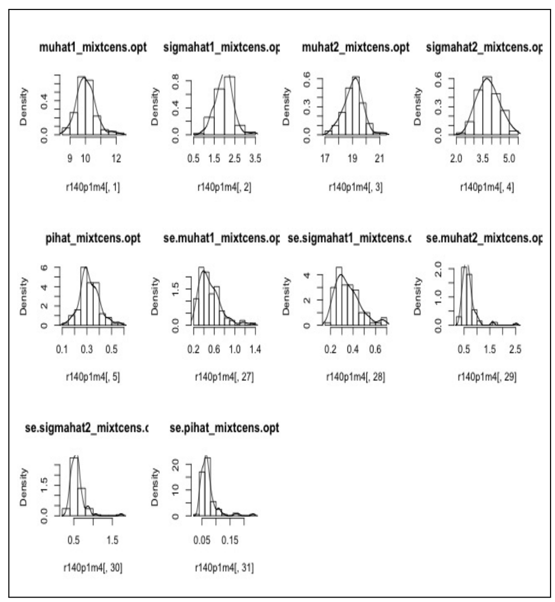


- when $r = 140$ and for parameter set i

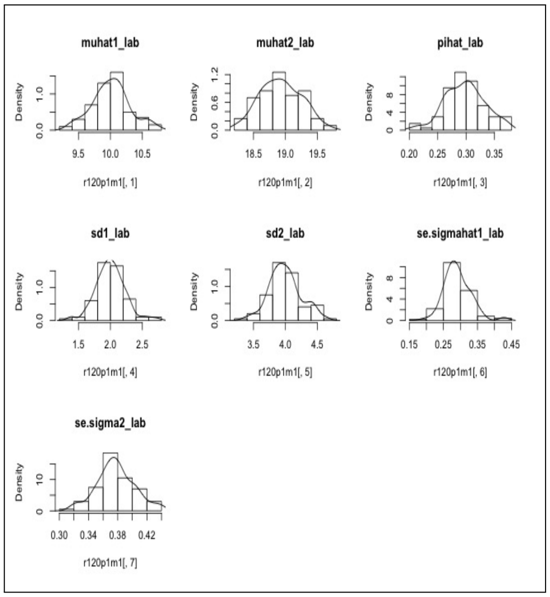


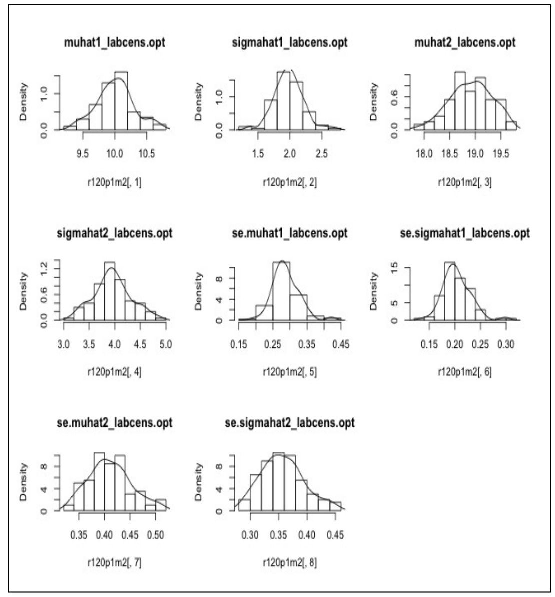


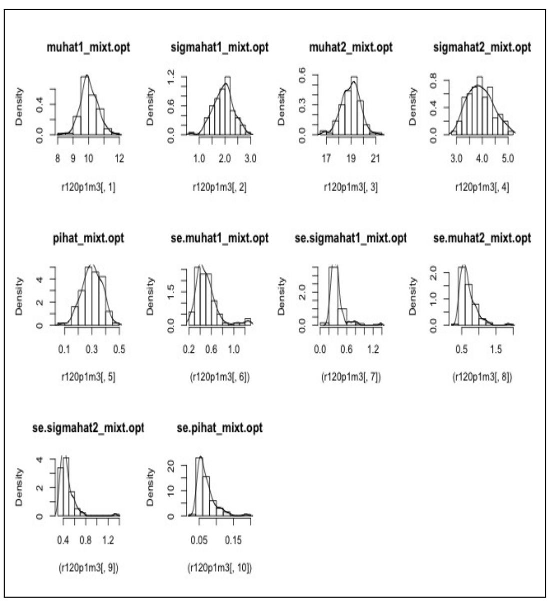


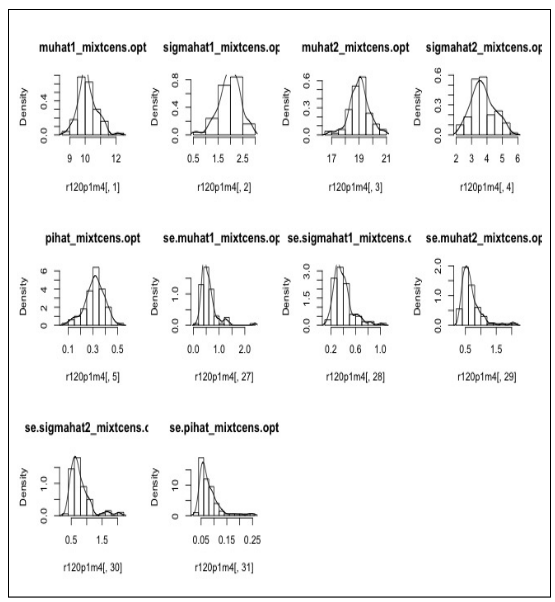


- when $r = 120$ and for parameter set i

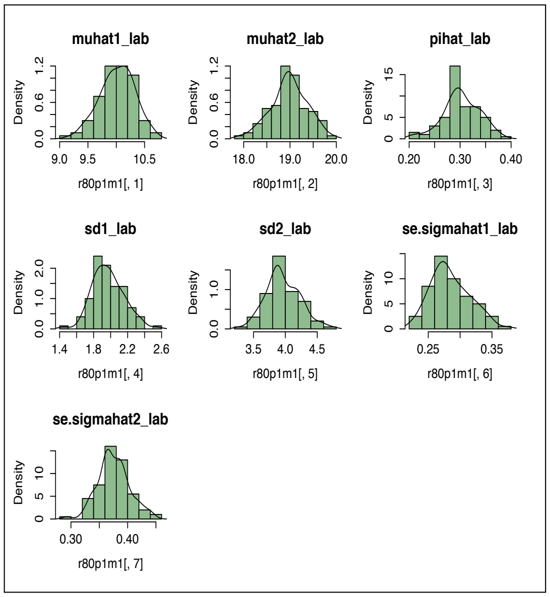


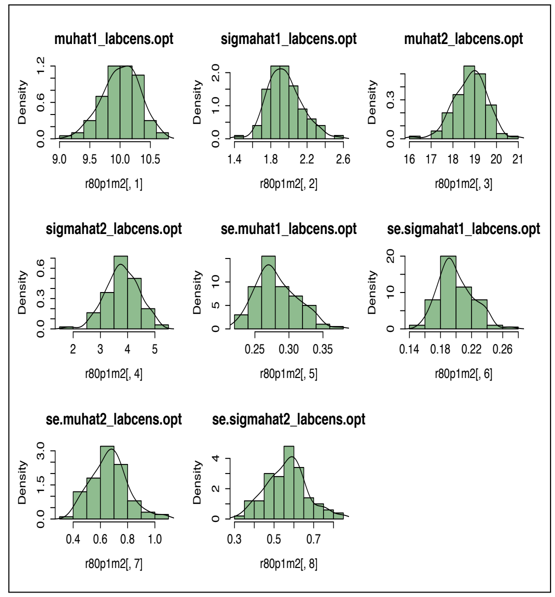


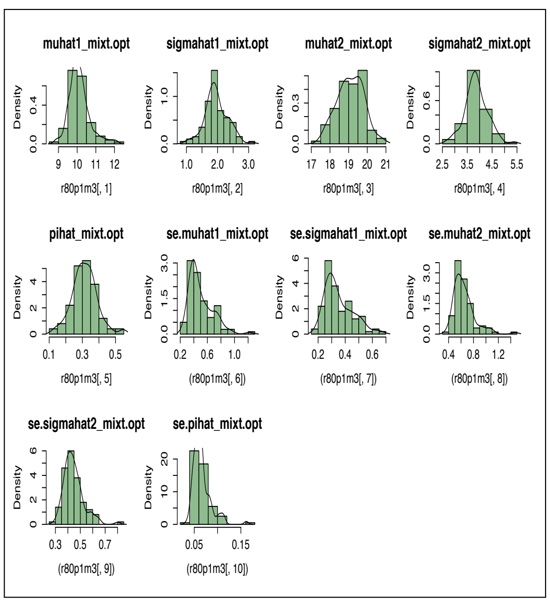


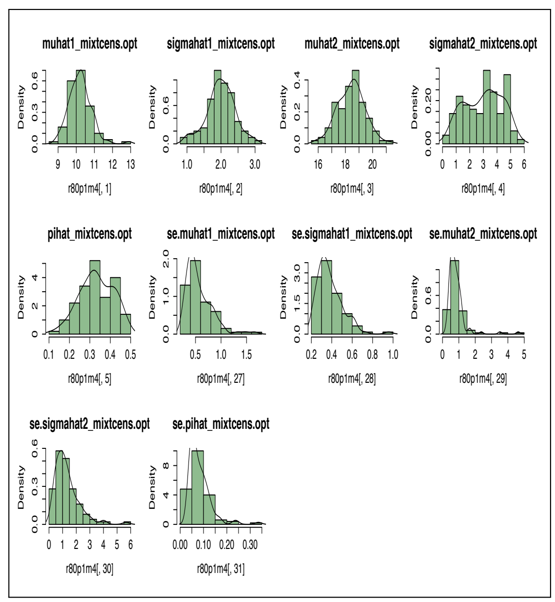


- when $r = 80$ and for parameter set i

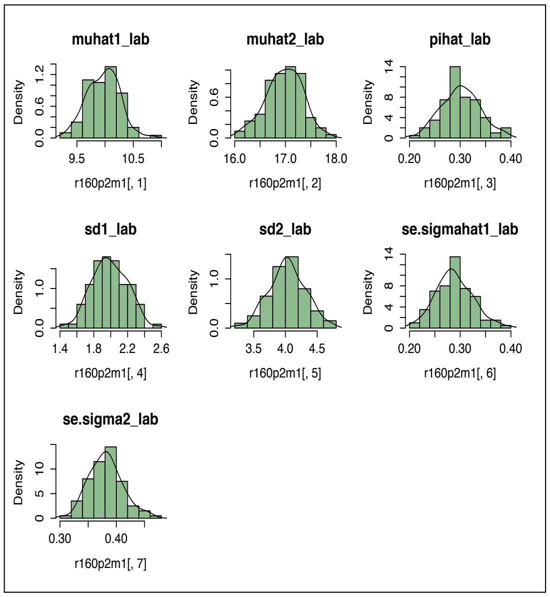


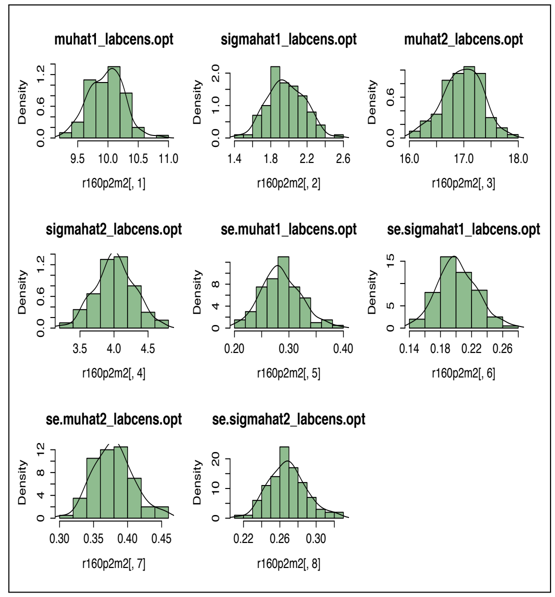


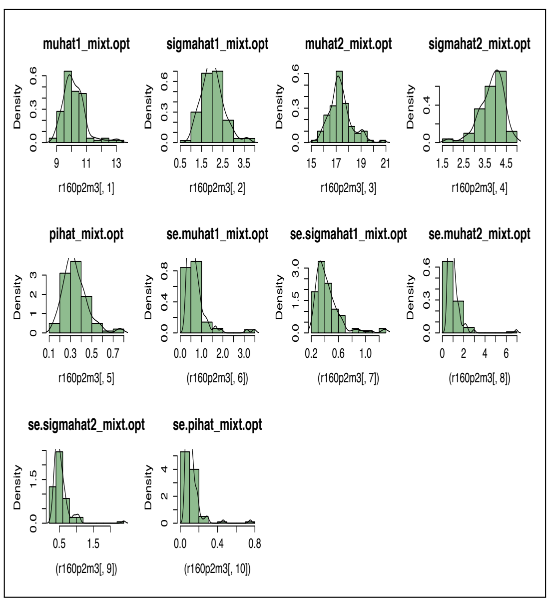


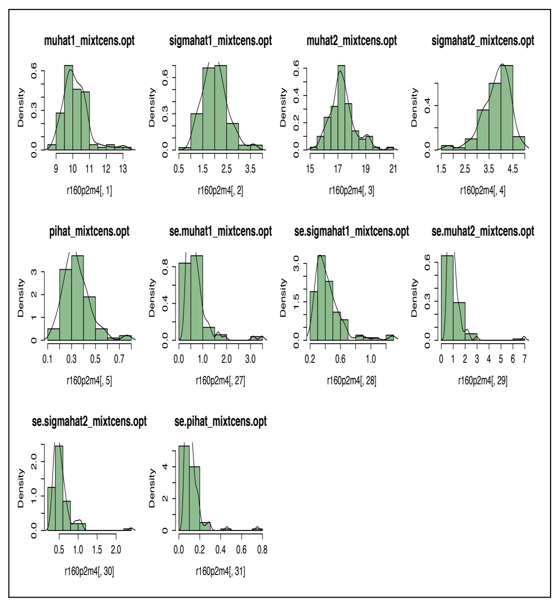


- when $r = 160$ and for parameter set ii

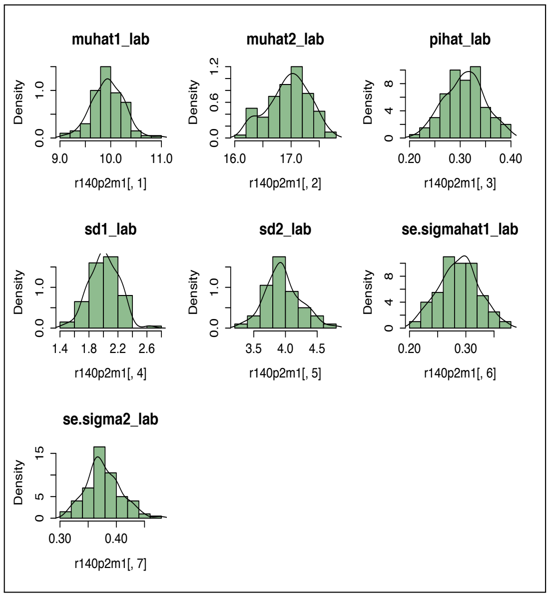


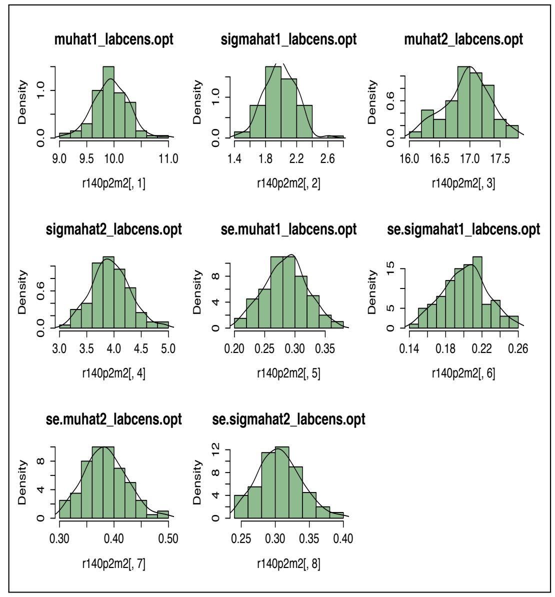


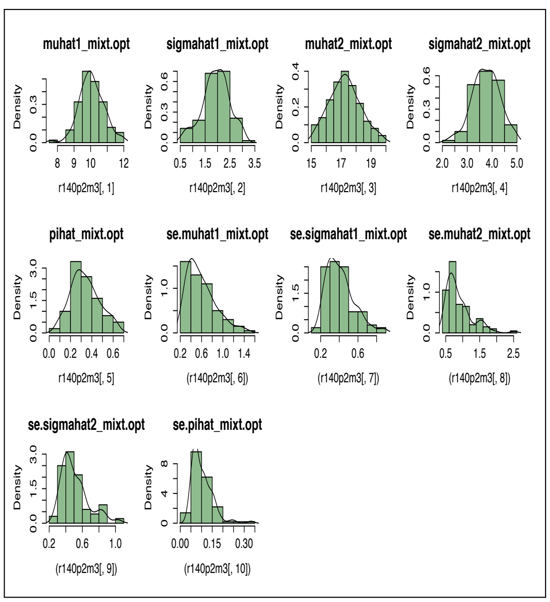


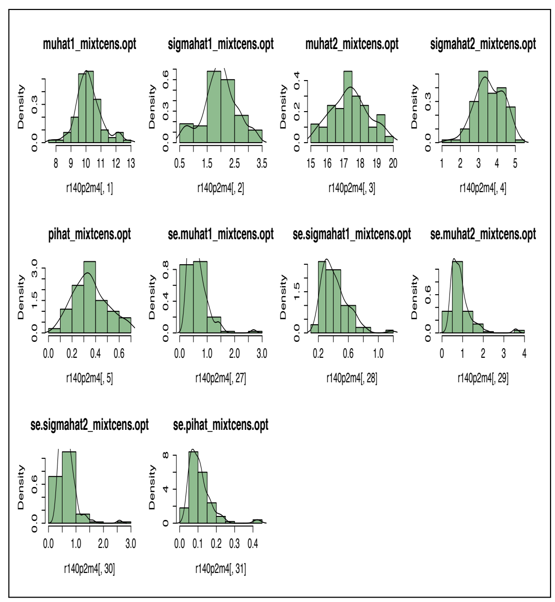


- when $r = 140$ and for parameter set ii

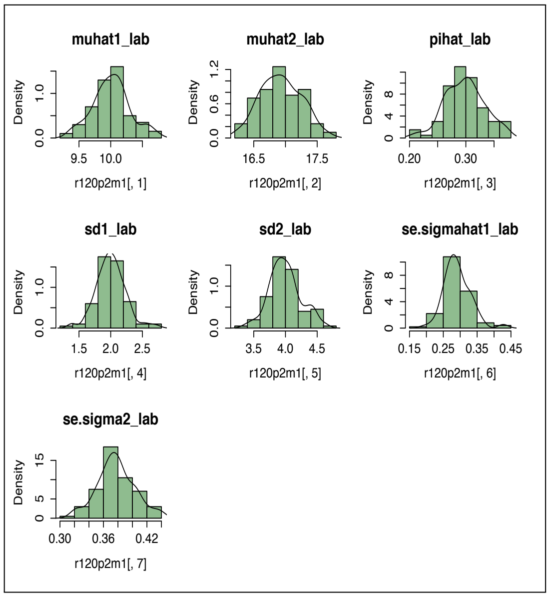


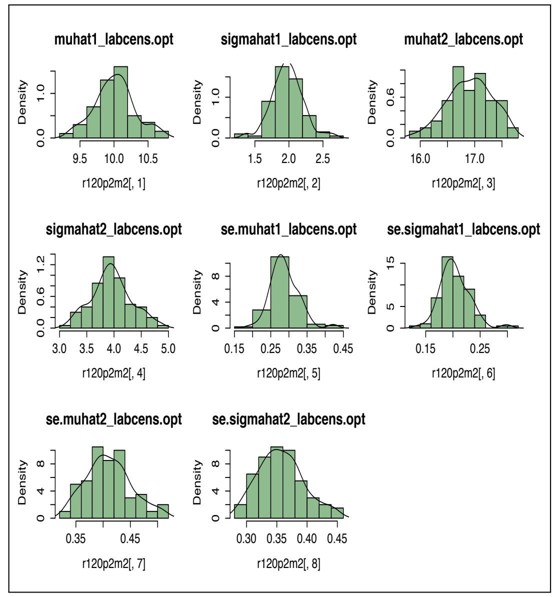


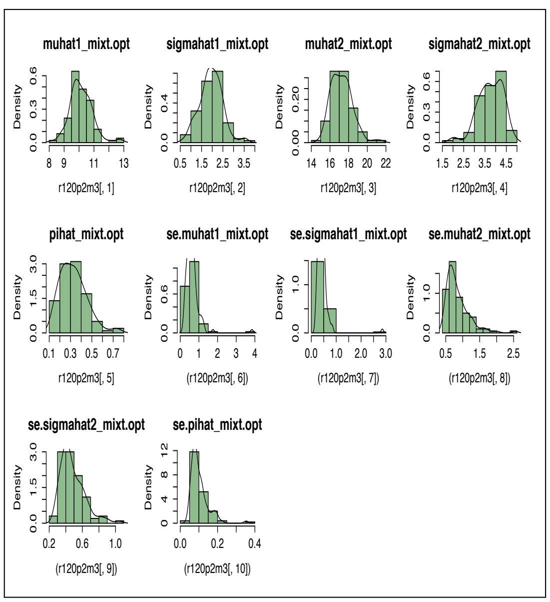


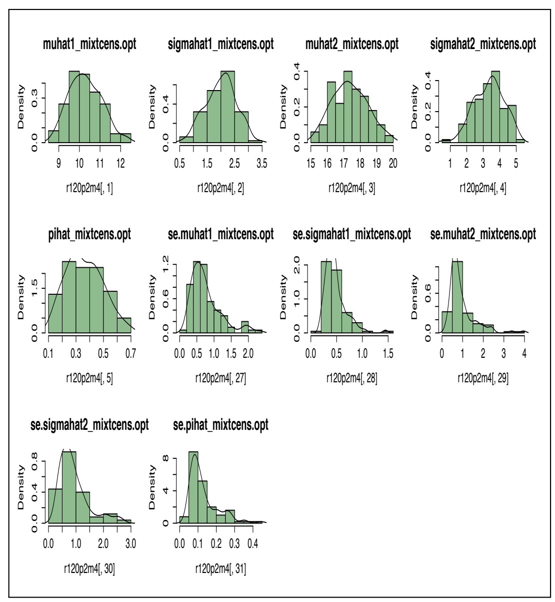


- when $r = 120$ and for parameter set ii

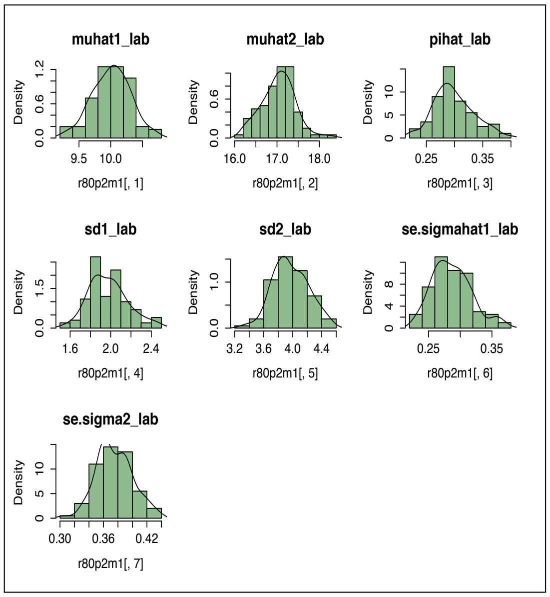


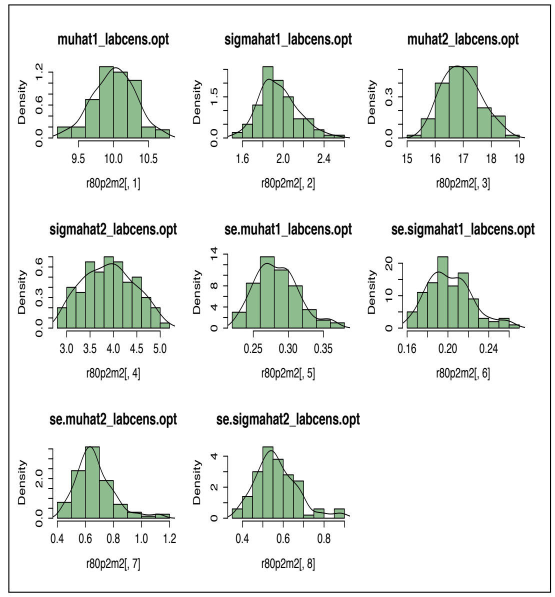


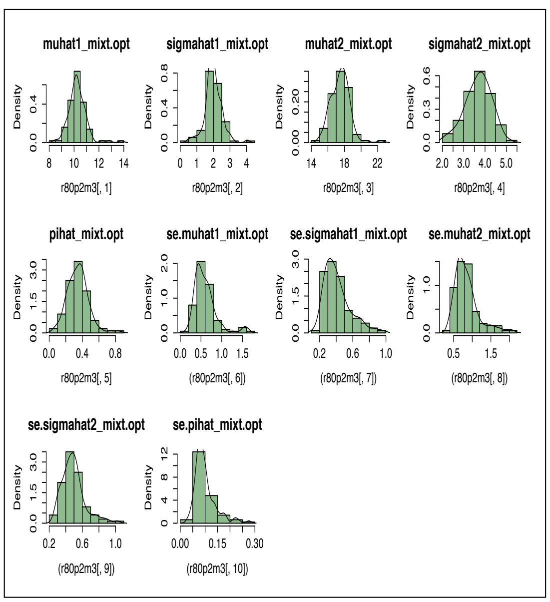


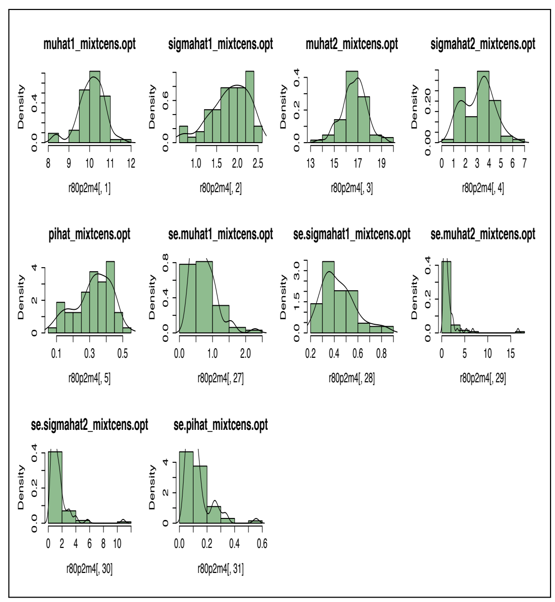


- when $r = 80$ and for parameter set ii









Chapter 4

Models for Longitudinal Clustered Survival Data

4.1 Introduction

Time to event, or survival, data are common in the agricultural, biological and medical sciences. Although time to an event is typically recorded as a continuous measurement, in some situations it can be discrete. For example, toxicology experiments may be designed such that treated insects are monitored with their mortality recorded only every couple of days. Or, in agricultural experiments on how some environmental conditions affect flowering, when controlling for other effects, as in Pipper et al. (2011), the plant flowering may only be recorded each week. These type of experiments result in datasets where the survival time responses are grouped over time intervals and we refer to these as discrete survival data. It is also called clustered survival data in the sense that by measuring at repeated discrete time points some clustering mechanism over the individuals is induced. Notice that these experiments can also be considered as longitudinal studies in which individuals are repeatedly measured over adjacent time intervals and interest focuses on whether an outcome (death, flowering, etc) has yet occurred.

Modeling the hazard function for this kind of clustered longitudinal survival data is typically done using the Generalized Linear Model (GLM) and Generalized Linear Mixed Model (GLMM), with appropriate link functions. These both allow for modeling non-normal responses (dependent variables), while with GLMM, correlated and uncorrelated random effects can be incorporated in the model to account for additional variation, often referred to as heterogeneity in outcome values. In GLMMs these additional components of variation are included in the linear predictor as random effects, which typically involves specification of the distribution of the random effects. A common assumption is to use a normal distribution. The distribution of random effects has been shown to be important in the estimation of parameters. For example, a misspecified random effect distribution, for either the marginal or joint distribution of a random intercept and slope, can lead to biased estimates of parameters and variance components and inaccurate predictions. McCulloch and Neuhaus (2011) and Neuhaus et al. (2013) have shown that misspecification of the joint distribution typically yields little bias in estimation of the fixed covariate effects. However, Litière et al. (2008) mentioned that the estimation of the random effects' variance components can be severely biased. An alternative way of fitting such models, is to leave the distribution of the random effects completely unspecified by considering a non-parametric distribution for them. The idea is that the distribution of random effects can be considered as an unknown finite mixture distribution, (see Einbeck and Hinde, 2006; Aitkin, 1996; Laird, 1978) and estimated along with the other model parameters. For more references on misspecified random effects, one could see Rizopoulos et al. (2008) or Verbeke and Molenberghs (2013).

Authors have analyzed clustered survival data in different alternative ways. For instance, Pipper et al. (2011) used a simple linear regression to model the hazard for such clustered data. Also Vaida and Xu (2000) used random effects for clusters as a generalized frailty term included into the Cox proportional hazards model formulation, forming a mixed effects model, and illustrated the application of their model to censored survival data. Here, we consider a dataset from a biological control assay with clustered responses, which is basically a collection of daily mortality measures of some insects exposed to certain different isolates of a microbial control. We fit repeated binomial models with a family of link functions for proportional and additive interval hazards, as introduced in Tibshirani and Ciampi (1983), to a random subset of this data. We then consider including random effects to account for additional variability within each longitudinal cluster in the context of GLMMs. Different hazard models are compared in terms of goodness of fit. Using a non-parametric random effect model with a finite mixture representation we consider using the induced clustering to find the most effective types of isolate.

4.2 The `insect` Dataset

The termite *Heterotermes tenuis* is a serious threat to sugarcane in Brazil, causing damage of up to 10 metric tones/ha/year, either during the planting season or the maturation phase of the cane, see Almeida et al. (1997). Recent research has looked at the use of biological control for this pest. Ongoing work at the Insect Pathology Laboratory of ESALQ-USP, Piracicaba, São Paulo, Brazil has focused on the use of the fungus *Beauveria bassiana* as a microbial control for *Heterotermes tenuis*. Experiments have been used to study the pathogenicity and virulence of different strains of the fungus with the aim of determining effective isolates for use in the field.

The dataset discussed in this chapter compares 142 different isolates of the fungus, using a completely randomized experiment with five replicates. Solutions of the isolates (108 particles/ml) were applied to groups (clusters) of $n = 30$ termites kept in plastic Petri-dishes (60 mm diameter 10 mm height), with five replicate dishes for each isolate. The mortality in the groups was measured by counting the dead termites daily for a period of eight days after application of the fungus, resulting in 5680 ($142 \times 8 \times 5$) repeated binomial observations (Note that it can also be thought of 710 (142×5) longitudinal multinomial observations of follow-up length eight). To get some idea about the nature of the response and components of variability, we start with a plot of data. Figure 4.1 shows the cumulative proportion of dead termites against day for a random sample of isolates with panels representing the isolates' labels.

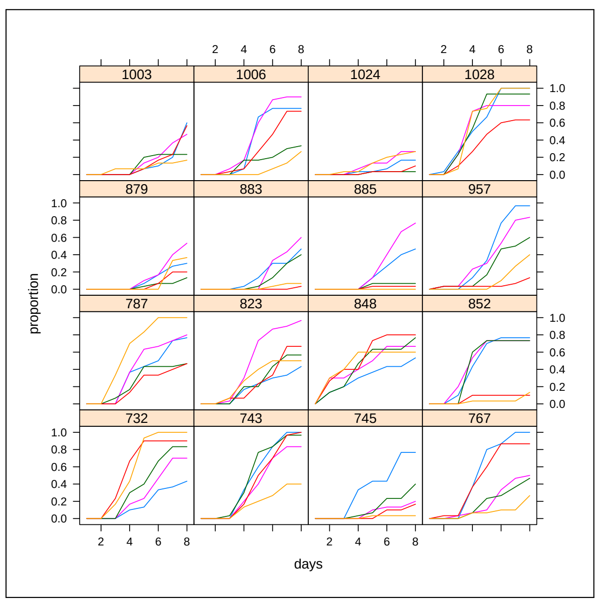


Figure 4.1: Cumulative mortality of termites against day

We see five replicates for each isolate with different death rates for each replicate and across isolates. It shows that there are clear differences in the efficacy of the isolates and evidence of differing degrees of variability among replicates within the isolates. This shows a model may need to include components to account for this isolate and replicate variability, as well as the cumulative mortality patterns over time, and random effects are a sensible approach. In the following, we model the cumulative mortality of these insects with and without random effects and compare the models.

The structure of the data is as given below.

```
> str(insect2)
'data.frame': 800 obs. of  6 variables:
$ isol   : Factor w/ 142 levels "282","309","353",..: 142 142 142 142 ...
$ r      : int  1 1 1 1 1 1 1 1 2 2 ...
$ day    : Factor w/ 8 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 1 2 ...
$ y      : int  0 0 0 0 4 4 4 2 0 0 ...
$ p      : num  0 0 0 0 0.133 ...
$ n      : int  30 30 30 30 30 30 30 30 30 30 ...
```

In this data frame, the response variable `p` is the observed proportion of death for the group of insects for the specific replicate of the isolate on a given day, while `y` denotes the observed number of deaths for this group, with `n` being the total number of insects at risk at the start of exposure to toxicity (application of fungus isolate) within each replicate. `isol` and `r` are factors representing the isolate and replicate within isolate, respectively, and `day` is the discrete time index.

4.3 Family of Models

We begin by considering a general framework for discrete survival data adapting the ideas from the continuous case. Suppose t_j, $j = 1, 2, \ldots, n$, are observed failure times of individuals and δ_j is a censoring indicator associated with them, where $\delta_j = 0$ if the j-th individual is censored. Also let $\boldsymbol{z}_j$ denote the covariate associated with individual j, which is assumed time-invariant. We denote the underlying continuous failure time distribution and survival function by $f(t; \boldsymbol{z})$ and $S(t; \boldsymbol{z})$ respectively. For such set-up one can write the hazard function as follows:

$$h(t; \boldsymbol{z}) = \lim_{\Delta t \to 0} \frac{P(t \leq T \leq t + \Delta t | T \geq t)}{\Delta t}$$

The proportional and additive hazards models are then defined simply as

$$h_j(t) = h_0(t) \exp(\boldsymbol{a}' \boldsymbol{z_j}) \tag{4.1}$$

and

$$h_j(t) = h_0(t) + \boldsymbol{a}' \boldsymbol{z_j} \tag{4.2}$$

respectively, where $\boldsymbol{a}$ denotes a vector of unknown parameters and $h_0(t)$ a baseline hazard (corresponding to an individual with $\boldsymbol{z}_j = 0$). Note that another standard assumption in the analysis of such data is that the distribution of the censoring times is independent of the failure times.

However, for discrete (grouped) survival data, defining the hazard model is not as simple. Suppose in an experimental study the individuals are monitored over time intervals $\mathcal{T}_i = [\tau_{i-1}, \tau_i)$, $i = 1, 2, \ldots, m$, where $\tau_0 = 0$ and τ_m is assumed to be any time after the study termination time. Then the interval hazard is defined by Tibshirani and Ciampi (1983) as follows:

$$p_i(\boldsymbol{z}) = Pr(\tau_{i-1} \leq T \leq \tau_i | T \geq \tau_{i-1}),$$

where p_i depends on the covariate $\boldsymbol{z}$ but does not vary over time within the interval, i.e., we assume that for a given $\boldsymbol{z}$, $p_i(\boldsymbol{z})$ is the piecewise constant hazard within the time interval $\mathcal{T}_i$.

It is simple to see that the piecewise hazard can be written in terms of the survival function as

$$p_i(\boldsymbol{z}) = Pr(\tau_{i-1} \leq T \leq \tau_i | T \geq \tau_{i-1}) = \frac{Pr(\tau_{i-1} \leq T \leq \tau_i)}{Pr(T \geq \tau_{i-1})} = 1 - \frac{S(\tau_i; \boldsymbol{z})}{S(\tau_{i-1}; \boldsymbol{z})} \tag{4.3}$$

and noting that

$$\begin{aligned} S(\tau_i; \boldsymbol{z}) &= \exp\left\{-\Lambda(\tau_i)\right\} \\ &= \exp\left[-\int_0^{\tau_i} h(s; \boldsymbol{z})\, ds\right], \end{aligned} \tag{4.4}$$

we can write the interval version of the hazard function as

$$\begin{aligned} 1 - p_i(\boldsymbol{z}) &= \frac{\exp\left[-\int_0^{\tau_i} h(s; \boldsymbol{z})\, ds\right]}{\exp\left[-\int_0^{\tau_{i-1}} h(s; \boldsymbol{z})\, ds\right]} \\ &= \exp\left[-\int_{\tau_{i-1}}^{\tau_i} h(s; \boldsymbol{z})\, ds\right]. \end{aligned} \tag{4.5}$$

Therefore, if we assume an underlying piecewise constant hazard on the collection of intervals $\mathcal{T}_i$, we can write

$$\begin{aligned} \ln(1 - p_i(\boldsymbol{z})) &= -\int_{\tau_{i-1}}^{\tau_i} h(s; \boldsymbol{z})\, ds \\ &= -(\tau_i - \tau_{i-1})\, h(\tau_{i-1}; \boldsymbol{z}) \end{aligned} \tag{4.6}$$

Now, we replace the multiplicative model (4.1), in (4.6) which results in the interval version of the proportional hazards model as follow.

$$\begin{aligned} -\ln(1-p_i(\boldsymbol{z})) &= (\tau_i - \tau_{i-1})\ h(\tau_{i-1};\boldsymbol{z}) \\ &= (\tau_i - \tau_{i-1})\ h(\tau_{i-1};\mathbf{0}) \exp\big(\boldsymbol{a}'\boldsymbol{z}\big) \\ &= -\ln\big(1-p_i(\mathbf{0})\big) \exp\big(\boldsymbol{a}'\boldsymbol{z}\big) \end{aligned}$$

Hence,

$$\ln\big[-\ln(1-p_i(\boldsymbol{z}))\big] = \ln\big[-\ln(1-p_i(\mathbf{0}))\big] + \boldsymbol{a}'\boldsymbol{z} \tag{4.7}$$

Also, if we replace model (4.2), in (4.6), we get the additive hazards model as obtained below.

$$\begin{aligned} \ln(1-p_i(\boldsymbol{z})) &= -(\tau_i - \tau_{i-1})\ h(\tau_{i-1};\boldsymbol{z}) \\ &= -(\tau_i - \tau_{i-1})\ \big[h(\tau_{i-1};\mathbf{0}) + \boldsymbol{a}'\boldsymbol{z}\big] \\ &= \ln\big(1-p_i(\mathbf{0})\big) - \boldsymbol{a}'\boldsymbol{z}\,(\tau_i - \tau_{i-1}) \end{aligned}$$

Therefore,

$$-\ln(1-p_i(\boldsymbol{z})) = -\ln(1-p_i(\mathbf{0})) + \boldsymbol{a}'\boldsymbol{z}\,(\tau_i - \tau_{i-1}) \tag{4.8}$$

Tibshirani and Ciampi (1983) wrote the interval version of the hazard models as in (4.7) and (4.8) and then generalized them to include polynomial time trends. Considering the time trend, the proportional hazards model and additive hazards model are defined respectively as

$$\ln\Big[-\ln\big\{1-p_i(\boldsymbol{z})\big\}\Big] = c_i + \sum_{j=1}^{s} \boldsymbol{a}_j'\boldsymbol{z}(\tau_i^j - \tau_{i-1}^j) \tag{4.9}$$

and

$$-\ln\big\{1-p_i(\boldsymbol{z})\big\} = c_i + \sum_{j=1}^{s} \boldsymbol{a}_j'\boldsymbol{z}(\tau_i^j - \tau_{i-1}^j) \tag{4.10}$$

where s denotes the degree of the polynomial time dependence and the $\{c_i\}$ are a set of parameters for the arbitrary baseline piecewise constant hazard. It can be seen that (4.9) and (4.10) show linear relationships between the covariate $\boldsymbol{z}$ and the transformed values of $p_i(\boldsymbol{z})$ under the relevant link functions.

In general, we can generate a family of hazard models by applying different link functions over $p_i(\boldsymbol{z})$. Using the parametric link function

$$V_\lambda(\pi) = \frac{1}{\lambda}\{[-\ln(1-\pi)]^\lambda - 1\}, \quad 0 \le \lambda \le 1 \tag{4.11}$$

introduced by Aranda-Ordaz (1981), this extended family of models is given as

$$V_\lambda(p_i(z)) = c_i + \sum_{j=1}^{s} \boldsymbol{a}'_j \boldsymbol{z}(\tau_i^j - \tau_{i-1}^j),$$

where c_i is the baseline hazard over the i-th time interval.

Notice that, $\lambda = 0$ leads in the limit to the complementary log-log link (`cloglog`) for $p_i(\boldsymbol{z})$ and thus corresponds to the proportional hazards model (4.9), while $\lambda = 1$ leads to the complementary log link (`clog`) and the additive hazards model (4.10). As these two models are embedded in the same larger parametric family of models, the deviances from fitting the models can be compared to choose between these additive and multiplicative hazard models. In principle the link parameter λ could be estimated as in Aranda-Ordaz (1981), however this is not pursued here.

Now, we will use these two models for modeling the insect mortality dataset with and without the inclusion of random effects and compare the results. For discrete survival data the outcome for each individual in each time interval is simply a binary response (death/survival), where survival in an interval corresponds to censoring in that interval while continuing to be at risk of death in the subsequent time intervals. In this way the likelihood function can be written as

$$L = \prod_{i=1}^{r=5} \prod_{j=1}^{\tau=8} \binom{n_{ij}}{f_{ij}} p_{ij}^{f_{ij}} (1 - p_{ij})^{n_{ij} - f_{ij}}, \tag{4.12}$$

which can be thought of as repeated binomial representation of a multinomial response over time for τ time intervals and r replicates, where f_{ij} is the number of failures/deaths for the i-th replicate in the j-th time interval (day), p_{ij} is the associated probability of death, and n_{ij} is the number at risk for the i-th replicate and on the j-th day. Given the nature of the grouped longitudinal observations process we have $n_{i,j+1} = n_{ij} - f_{ij}$, that is the number at risk on day $j + 1$ is the number at risk on the previous day minus the number of deaths.

To study the properties of the model fitting procedure, we consider a random sample of 20 isolates and fit logistic regression models in a GLM structure using the `glm` function with link functions as specified in (4.11). We then consider the inclusion of random effects in a GLMM structure using the `glmer` function in `R`.

4.3.1 Logistic regression model for proportional hazards

As discussed above, we consider a logistic model with `cloglog` link for fitting a proportional hazards model; this link function is already defined for fitting models from a `binomial` family with `glm`. We include a factor "day" for the discrete time intervals to give the baseline hazard and consider an additive time-invariant isolate effect. The model is formulated as

$$\begin{aligned} E\left(y_{ij}\right) &= \mu_{ij}, \\ g(\mu_{ij}) &= \log\left[-\log(1-\mu_{ij})\right] \\ &= \boldsymbol{x}_i' \boldsymbol{\beta} \quad i=1,\ldots,20; j=1,\ldots,5, \end{aligned} \tag{4.13}$$

where $\boldsymbol{x}_i'$ is the i-th row of the design matrix for the fixed effects, $\boldsymbol{\beta}$ is the vector of fixed effect coefficients for day and for the set of isolates represented by the factor `isolate`. This model is fitted in `R` as below.

```
R> m1 <- glm(cbind(y, n_risk - y) ~ -1 + day + isol, family =

               binomial(link = cloglog), data = insect2)          (4.14)
```

Note 4.1. *Here, and throughout this chapter, the variables* ***y*** *and* ***n_risk*** *correspond to* f_{ij} *and* n_{ij} *in (4.12), respectively,*

The following gives the summary output of the model:

```
R> summary(m1)

Call:
glm(formula = cbind(y, n_risk - y) ~ -1 + day + isol,
                     family = binomial(link = cloglog), data = insect2)

Deviance Residuals:
Min       1Q   Median      3Q      Max
-4.4620  -1.4451  -0.8801   0.8106   6.7973

Coefficients:
        Estimate    Std. Error z value Pr(>|z|)
day1    -3.9329548  0.1621609 -24.253  < 2e-16 ***
day2    -3.7061510  0.1525162 -24.300  < 2e-16 ***
```

```
day3    -2.7922467  0.1233608 -22.635  < 2e-16 ***
day4    -1.3189827  0.1044254 -12.631  < 2e-16 ***
day5    -1.1361904  0.1058834 -10.731  < 2e-16 ***
day6    -1.2088743  0.1111702 -10.874  < 2e-16 ***
day7    -1.9105503  0.1294749 -14.756  < 2e-16 ***
day8    -2.3772698  0.1491824 -15.935  < 2e-16 ***
isol309 -0.3871173  0.1466777  -2.639 0.008309 **
isol353 -0.8878904  0.1602402  -5.541 3.01e-08 ***
isol489  0.3792722  0.1369517   2.769 0.005616 **
isol494  0.5482638  0.1330489   4.121 3.78e-05 ***
isol500  0.6182967  0.1334222   4.634 3.58e-06 ***
isol524  0.4166775  0.1334924   3.121 0.001800 **
isol534  0.0002618  0.1365259   0.002 0.998470
isol548  0.0330131  0.1382266   0.239 0.811235
isol549 -0.2867903  0.1421627  -2.017 0.043660 *
isol569 -0.5874050  0.1520819  -3.862 0.000112 ***
isol571 -0.3946500  0.1552501  -2.542 0.011021 *
isol612  0.1578165  0.1380980   1.143 0.253127
isol665  0.0468823  0.1388841   0.338 0.735692
isol764  0.4677272  0.1330306   3.516 0.000438 ***
isol778  0.2158359  0.1331775   1.621 0.105090
isol783 -0.4567568  0.1463722  -3.121 0.001805 **
isol784 -0.6437051  0.1542969  -4.172 3.02e-05 ***
isol845  0.2117850  0.1348723   1.570 0.116354
isol885 -1.4488087  0.1868368  -7.754 8.88e-15 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 25271.7  on 778  degrees of freedom
Residual deviance:  2948.3  on 751  degrees of freedom
AIC: 4118.6

Number of Fisher Scoring iterations: 5
```

This model accounts for effects of within isolate groups of observations. This means that fitted values for observations within different isolates are influenced by the specific isolates and if we were to fit a model at the population level with a common hazard, i.e. with the effect of `day` only, we will be ignoring important isolate variation. This difference between the fitted values of these two models can be seen in Figure 4.2.

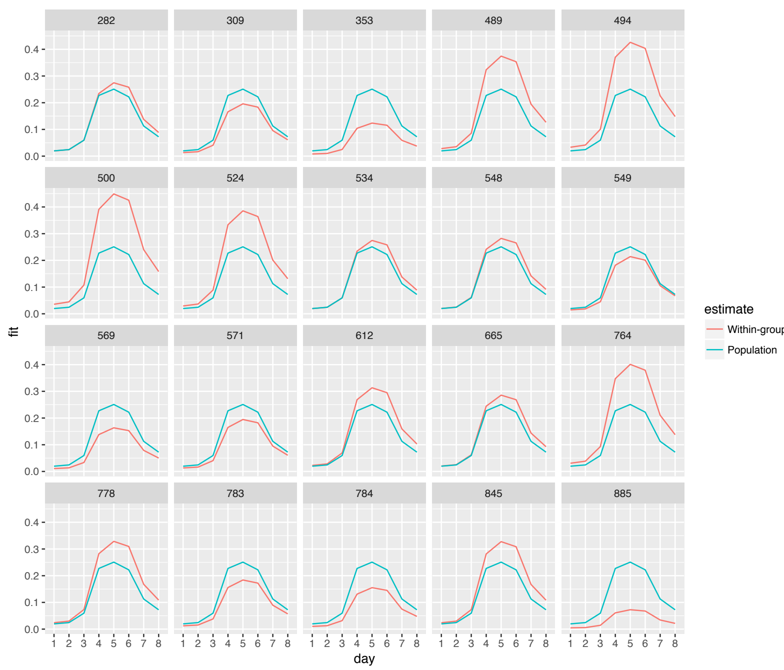


Figure 4.2: Fitted values estimated at the overall population level and with the additional isolate-group effect

In order to illustrate the different behaviour of isolates on the termites, we plot the fitted coefficients for the isolates given by model (4.14). This is shown in Figure 4.3, where the randomly sampled isolates are the labels on the x-axis and the y-axis values are their associated coefficient values for the mortality rate on the linear predictor scale. The isolates having similar behaviour are clustered visually with polygons with different colours to suggest possible different clusters of similar isolates. One can see that isolate 885 has a completely different behaviour, which marks it out as a possible outlier. We will re-look at this later when fitting random effect models.

Converting the estimates of the baseline hazard back to the original scale, using the inverse link expression $1-\exp(-\exp(\cdot))$, we can plot the baseline piecewise hazard function. This is shown in Figure 4.4. The model goodness-of-fit is evaluated visually in Figure 4.5, where we can see that the model does indeed pick up the general cumulative mortality trend and also reflects some of the variability between isolates. We see that isolate 885 is indeed rather different from the others, with generally lower mortality but also appreciably greater variation between replicates.

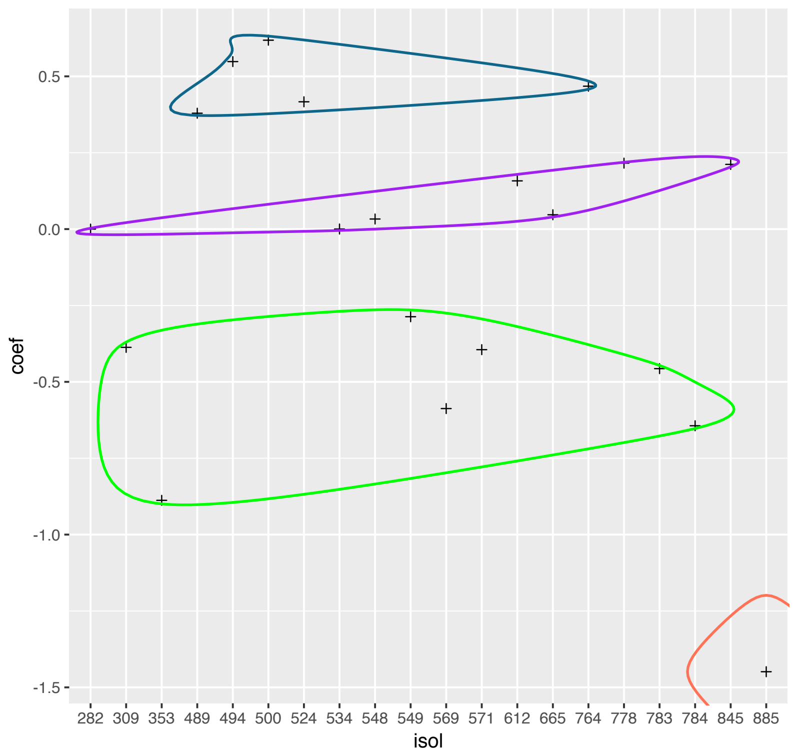


Figure 4.3: Clusters of isolates according to the estimated coefficient values of model (4.14)

4.3.2 Logistic regression model for additive hazards

As noted above, an additive hazards model for the discrete survival data can be fitted using the `clog` link. However, this link function is not already defined for the `binomial` family of `glm`, hence we have to define the link and associated functions for the `glm` fitting algorithm.

Several components need to be defined in `R` to create a new link function for a generalized liner model. These are listed below:

- link function g: given by $\eta = g(\mu) = -\log(1-\mu)$, where μ is the mean and η the linear predictor.

- inverse link function g^{-1}: given by solving the above equation for μ as

$$\exp\{-\eta\} = 1 - \mu, \tag{4.15}$$

  therefore the inverse link is given by $g^{-1}(\eta) = 1 - e^{\eta}$.

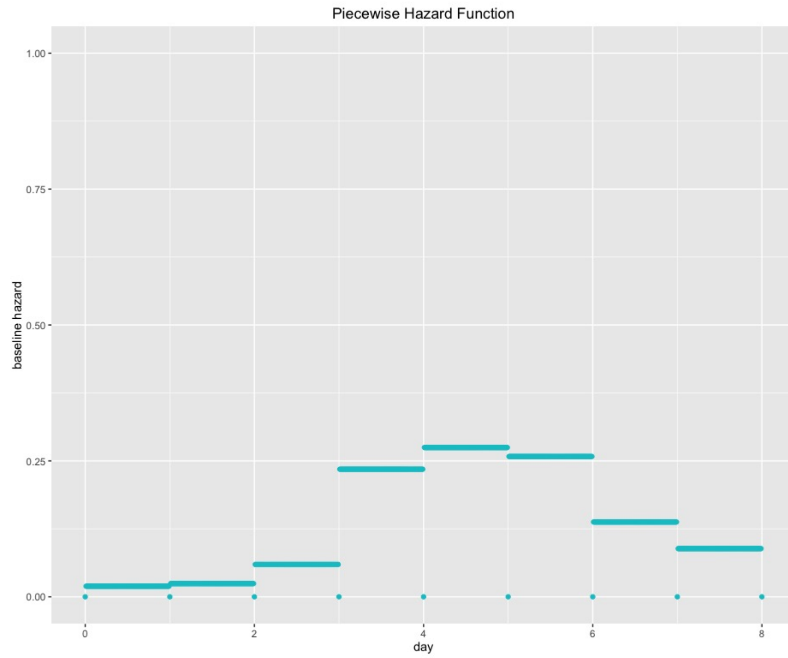


Figure 4.4: Baseline hazard function estimate based on model (4.14)

- link derivative $\frac{d\mu}{d\eta}$: solving equation (4.15) for μ and differentiating gives

$$\frac{d\mu}{d\eta} = e^{-\eta}.$$

The above components are gathered together to form a new link function in R as follows.

```
> clog <- function(mu,eta)
+ {
+   linkfun <- function(mu) -log(1-mu)
+   linkinv <- function(eta) 1-exp(-eta)
+   mu.eta <- function(eta) exp(-eta)
+   valideta <- function(eta) all(eta > 0)
+   link <- paste0("clog()")
+   structure(list(linkfun = linkfun, linkinv = linkinv,
+                  mu.eta = mu.eta, valideta = valideta, name = link),
+             class = "link-glm")
+ }
```

where `linkfun` transforms/links the mean of the response to the linear predictor η, that is it relates the mean to a linear model through a link function; `linkinv` is the inverse of

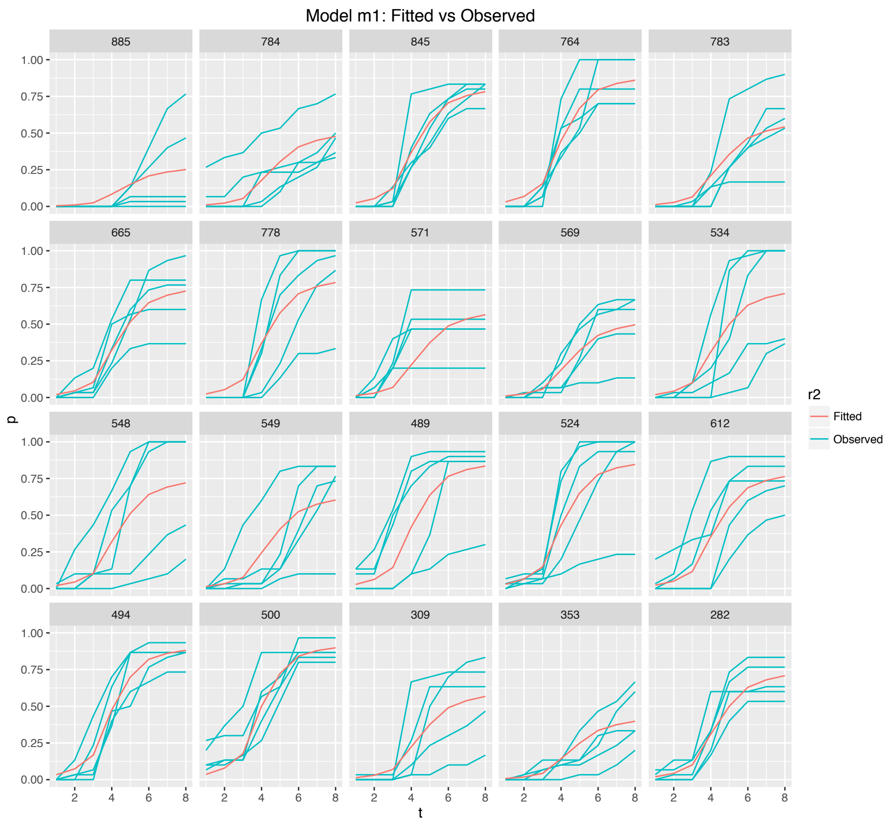


Figure 4.5: Fitted versus observed values based on model (4.14)

the link function `linkfun` which is required for transforming back the estimates or fitted values to the original/response scale; `mu.eta` is the derivative of μ with respect to η; and `valideta` is a logical function returning `TRUE` if the linear predictor η is within the domain of `linkinv`.

Once defined, we can include this link in the `binomial` family for later usage as follows.

```
> clog<-clog(mu,eta)
> bin_clog <- binomial()
> for (i in names(clog))
+   bin_clog[[i]] <- clog[[i]]
>
> binomial(clog)

Family: binomial
Link function: clog()
```

Hence, we can now fit a similar model to (4.14), except using the `clog` link defined above and compare the models in sense of their goodness of fit, with the `R` code as in (4.16). Note that the mathematical formulation of the following model fit is similar to (4.13) but with the different link function $g(\mu_{ij}) = -\log(1 - \mu_{ij})$.

```
R> m2 <- glm(cbind(y, n_risk - y) ~ -1 + day + isol, family =

             binomial(link = clog), start = exp(coef(m1)), data = insect2)
```
$$(4.16)$$

One minor difference here is that we need to supply appropriate starting values, and in order to ensure that starting values used by `glm` model are in the domain of the `clog` link function, we put the starting values equal to the exponential of the coefficients given by model (4.14). A summary of the model is given below.

```
R> summary(m2)

Call:
glm(formula = cbind(y, n_risk - y) ~ -1 + day + isol, family = binomial(link = clog),
data = insect2, start = exp(coef(m1)))

Deviance Residuals:
Min       1Q   Median       3Q      Max
-4.0899  -1.6118  -0.6834   0.6766   7.2497

Coefficients:
         Estimate   Std. Error z value Pr(>|z|)
day1      1.819e-02  6.991e-03   2.601 0.009290 **
day2      1.893e-02  7.006e-03   2.702 0.006887 **
day3      4.426e-02  7.833e-03   5.650 1.60e-08 ***
day4      2.372e-01  1.222e-02  19.410  < 2e-16 ***
day5      2.773e-01  1.427e-02  19.436  < 2e-16 ***
day6      2.367e-01  1.489e-02  15.892  < 2e-16 ***
day7      1.202e-01  1.234e-02   9.739  < 2e-16 ***
day8      7.866e-02  1.102e-02   7.140 9.30e-13 ***
isol309 -1.527e-02  7.637e-03  -1.999 0.045592 *
isol353 -1.308e-02  7.978e-03  -1.640 0.100970
isol489  7.117e-02  1.554e-02   4.580 4.65e-06 ***
isol494  5.222e-02  1.426e-02   3.662 0.000251 ***
isol500  9.631e-02  1.763e-02   5.463 4.69e-08 ***
```

```
isol524  3.385e-02  1.294e-02   2.617 0.008882 **
isol534 -1.004e-03  9.685e-03  -0.104 0.917455
isol548  2.651e-02  1.214e-02   2.183 0.029018 *
isol549  1.019e-02  1.066e-02   0.956 0.339081
isol569 -1.521e-02  7.641e-03  -1.990 0.046543 *
isol571  1.827e-02  1.135e-02   1.609 0.107590
isol612  4.157e-02  1.338e-02   3.107 0.001889 **
isol665  6.675e-03  1.047e-02   0.637 0.523831
isol764  2.436e-02  1.219e-02   1.999 0.045616 *
isol778  8.705e-05  9.883e-03   0.009 0.992972
isol783 -1.741e-02  7.218e-03  -2.412 0.015847 *
isol784  6.788e-03  1.034e-02   0.657 0.511323
isol845  6.849e-03  1.055e-02   0.649 0.516394
isol885 -1.819e-02  6.991e-03  -2.601 0.009290 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

Null deviance:    Inf  on 778  degrees of freedom
Residual deviance: 3184.9  on 751  degrees of freedom
AIC: 4355.1

Number of Fisher Scoring iterations: 25
```

The goodness of fit is visualized in Figure 4.6 by plotting the observed and fitted lines of cumulative mortality in the same panel. As can be seen, the observed cumulative mortality curves are again quite similar to the fitted lines.

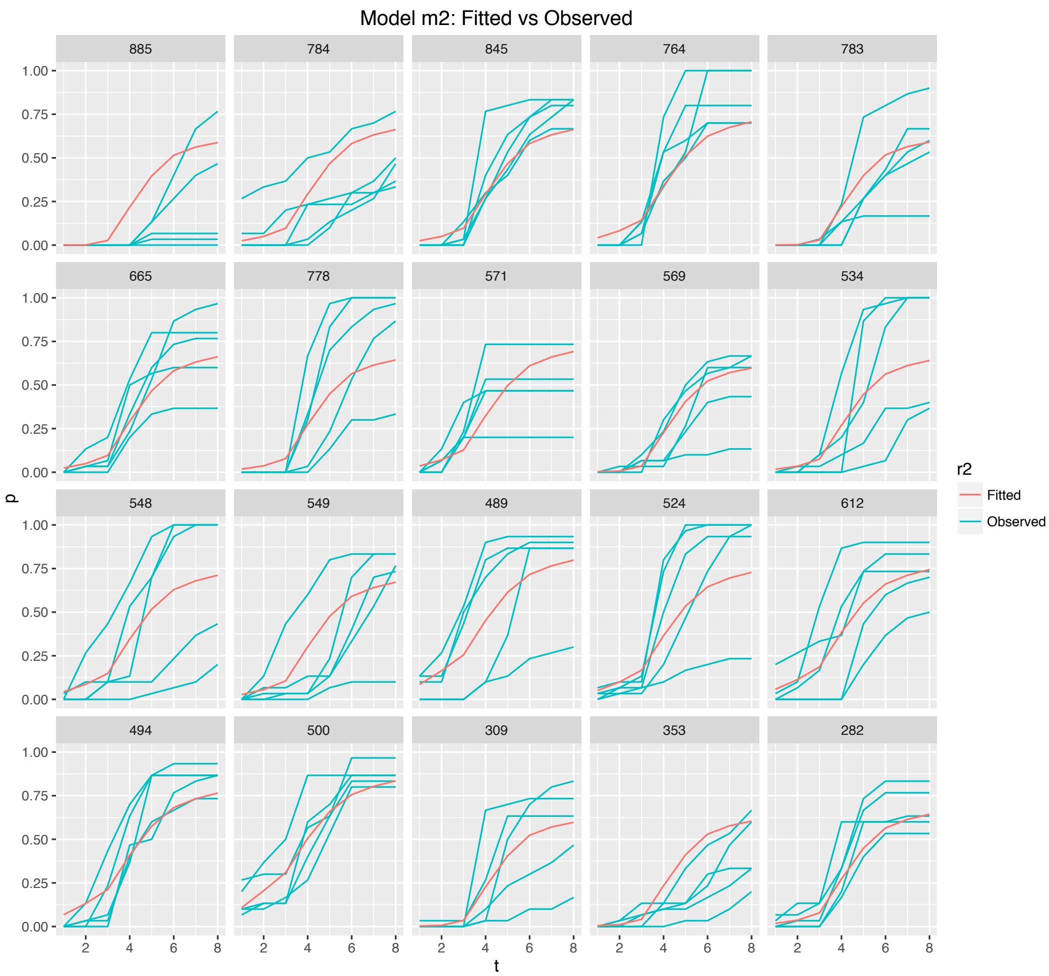


Figure 4.6: Cumulative Mortality

Comparing the results from the additive model (4.16) to the multiplicative model (4.14) indicates the better fit of the latter model since its residual deviance is lower with the same degrees of freedom, as we have the same terms in the linear predictor. We look at a plot of the isolate estimates from these two models against each other to illustrate their similarity. Figure 4.7 shows that these estimates for the isolates are of course numerically different, being on different scales, but that there is some general agreement between them. For example, the estimate for the isolate 885 is low in both models, while the highest estimate on both models is for isolate 500. Furthermore, from Figure 4.8 we see that the baseline hazard estimates from these models have similar patterns, although are not exactly the same numerically. This relationship can also be seen in Figure 4.9 which pictures the piecewise baseline hazard estimates from each model side by side. It is clear that the hazard is initially low, then elevated from day 3 to day 6 with subsequent reduction until the end of the observation period.

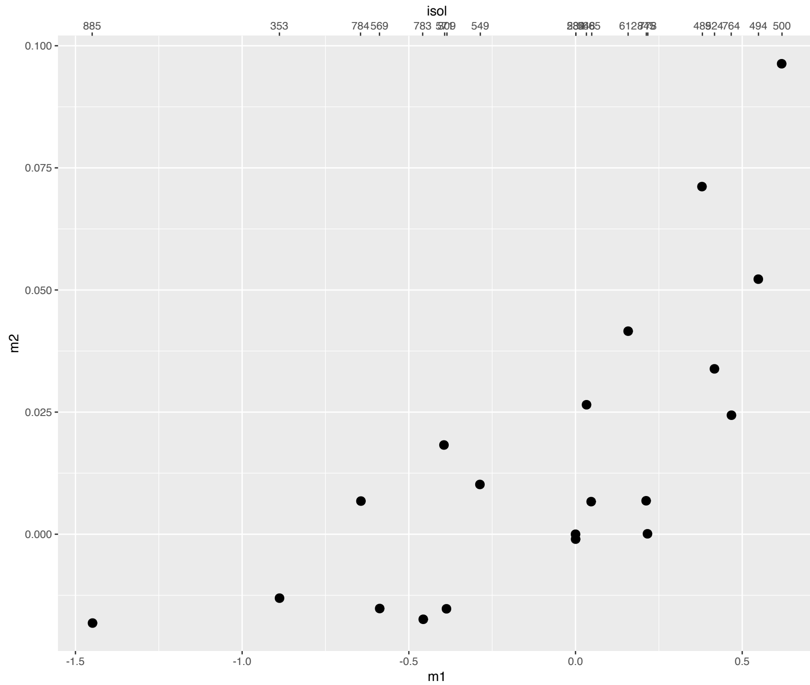


Figure 4.7: Plot of estimated coefficients for isolates from models `m1` and `m2` against each other

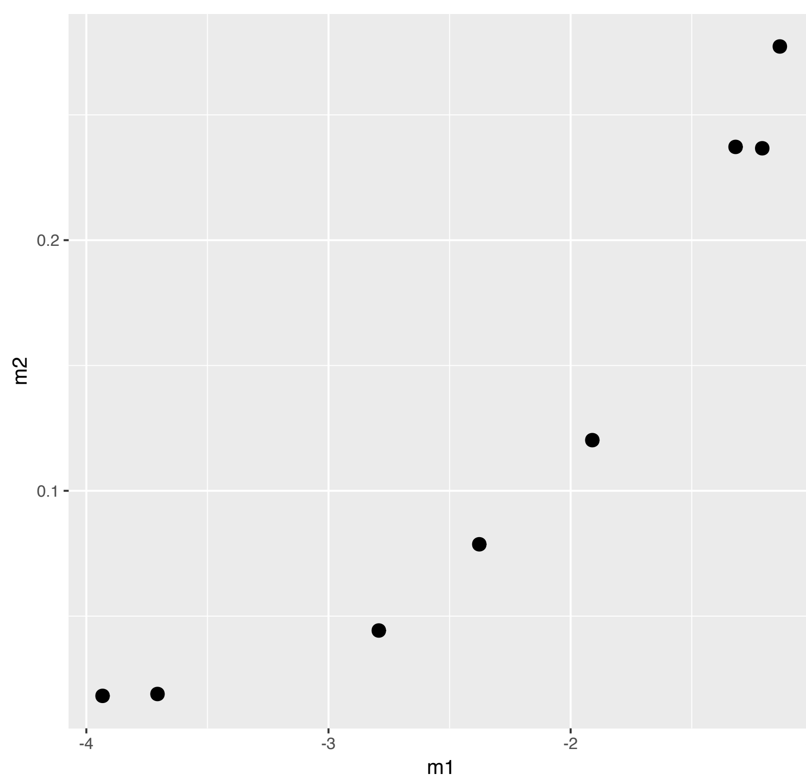


Figure 4.8: Plot of estimated coefficients for baseline hazards from models `m1` and `m2` against each other

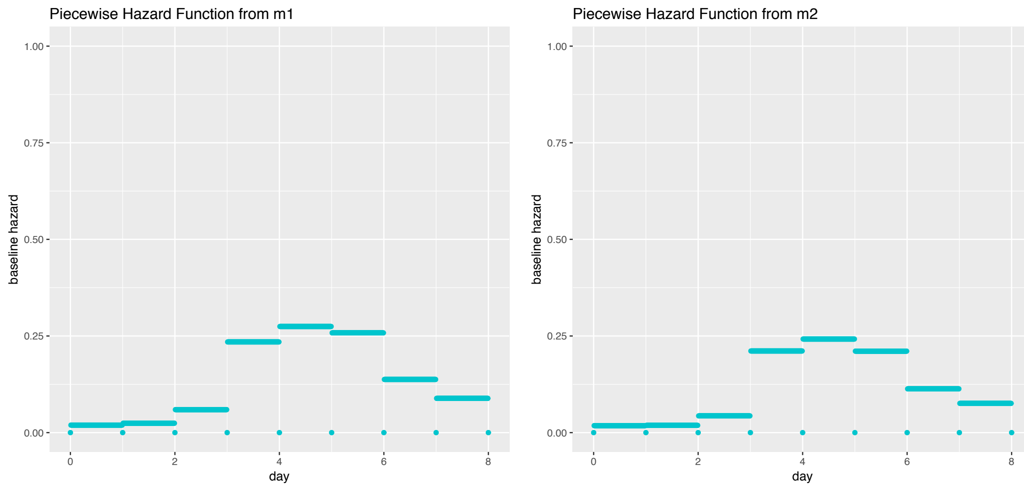


Figure 4.9: Plot of estimated coefficients for baseline hazards from models `m1` on the left and `m2` on the right

4.3.3 Proportional hazards models with random effects

As the effects of isolates on the insects' mortality varies and the set of specific isolates can be considered as a sample of the population of possible isolates of *Beauveria bassiana*, it makes sense to treat the isolates as a random factor. Also there is evidence of considerable variation between the replicates within each isolate, see Figure 4.1. Notice that the factor `day` is a fixed effect covariate that provides estimates for the baseline hazard and so to incorporate additional components of variation we need to consider fitting mixed effect models. To explore and find the best fit to this data, we consider adding two random effect terms one at a time and check the improvement in the model.

One random effect term

Here we fit a generalized linear mixed model with `day` as a fixed effect and `isol` as a random term using the following formulation in `R`.

```
ran.isol <- glmer(cbind(y, n_risk - y) ~ -1 + day + (1 | isol), data = insect2,
                  family = binomial(link = cloglog)),
```

where `(1 | isol)` is a simple scalar random effect for isolate (a random intercept for the cumulative mortality profiles). The summary output of the model is given below.

```
> summary(ran.isol)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation)
['glmerMod']
Family: binomial  ( cloglog )
```

```
Formula: cbind(y, n_risk - y) ~ -1 + day + (1 | isol)
Data: insect2

AIC      BIC     logLik    deviance    df.resid
4167.4   4209.6  -2074.7   4149.4      791

Scaled residuals:
Min      1Q  Median      3Q     Max
-3.5803 -1.0950 -0.6409  0.8835 13.7159

Random effects:
Groups Name        Variance Std.Dev.
isol   (Intercept) 0.2514   0.5014
Number of obs: 800, groups:  isol, 20

Fixed effects:
      Estimate  Std. Error z value Pr(>|z|)
day1  -4.0270     0.1720  -23.42   <2e-16 ***
day2  -3.8004     0.1628  -23.34   <2e-16 ***
day3  -2.8866     0.1360  -21.22   <2e-16 ***
day4  -1.4144     0.1195  -11.83   <2e-16 ***
day5  -1.2343     0.1206  -10.24   <2e-16 ***
day6  -1.3100     0.1244  -10.53   <2e-16 ***
day7  -2.0138     0.1409  -14.29   <2e-16 ***
day8  -2.4806     0.1592  -15.58   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The plot of isolate random effects from this model is shown in Figure 4.10. As we can see it again suggests four clusters, and in comparison with Figure 4.3 the same isolates are grouped within the same clusters. This demonstrates that the similarity between the random and fixed effect isolate models and so to account for the overdispersion apparent with these data another source of variability needs to be added to the model. This is explored in more detail in the next sections where we fit other mixed effects models, with two random effect terms and also with a non-parametric random effects distribution. We compare these with the current model in terms of their goodness of fit and model behaviour.

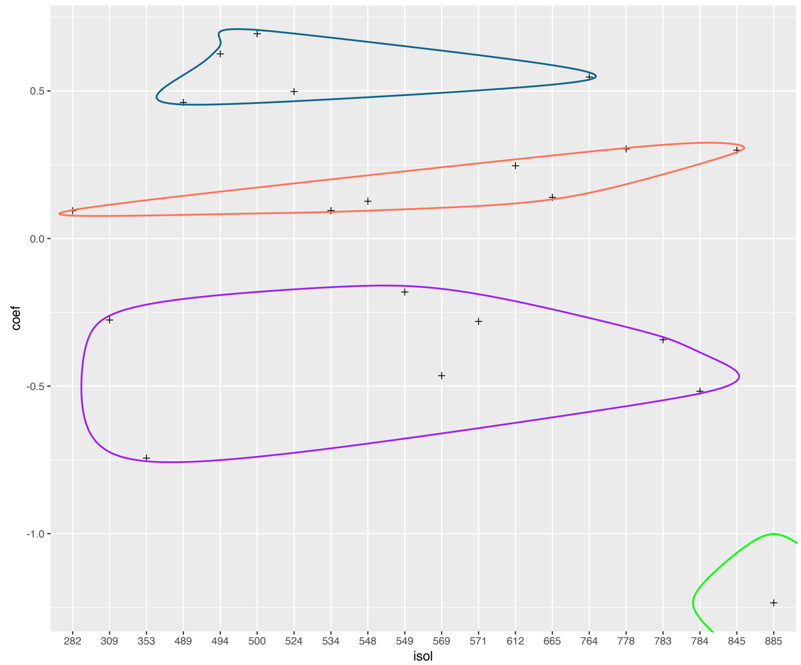


Figure 4.10: Clusters of isolates according to the model with one random effect term only

Two random effect terms

Here, we include two random effect terms in the model, (i) a simple scalar random effect for isolates (random intercept), and (ii) a random effect for the replicates also indexed by the isolates. The latter is referred to as nested random effect of replicates within isolates and is included in the model to account for within-isolate between-replicate variation. We define a GLMM for this data in the following way,

$$\begin{aligned} E\left(y_{ijk}|\boldsymbol{b},\boldsymbol{c}\right) &= \mu_{ijk}, \\ g(\mu_{ijk}) &= \log\left[-\log(1-\mu_{ijk})\right] \\ &= \beta_k + b_i + c_{ij} \quad i=1,\ldots,20; j=1,\ldots,5; k=1,\ldots,8, \end{aligned} \tag{4.17}$$

where y_{ijk} indicates the k-th day observation for the i-th isolate and j-th replicate. In (4.17), $\{\beta_k\}$ are the fixed effects for day, b_i is the random effect of the i-th isolate and c_{ij} is the nested random effect of replicate j within isolate i. $\boldsymbol{b} = (b_1, \ldots, b_{20})$ is assumed to follow a $N(\mathbf{0}, \sigma^2 I)$ distribution and $\boldsymbol{c} = (c_{11}, \ldots, c_{20,5})$ a $N(\mathbf{0}, \tau^2 I)$, where the variances σ^2 and τ^2 are unknown.

This model is formulated in `R` as follows: the syntax `(1 | isol)` represents a simple scalar random effect for isolate (random intercept) and a random intercept effect with respect to replicates within isolates is indicated by `(1 | isol:r)`. The sum of these two terms in the model on the right hand side of `glmer` function can be denoted for short by `(1 | isol/r)`, which means that the intercept varies among isolates and among replicates within isolates, i.e., the nested effect of replicates within isolates. For more information on the syntax for a GLMM, see Bates et al. (2015) and Bolker et al. (2017).

$$\text{ran.rep} <- \text{glmer(cbind(y, n\_risk} - \text{y)} \,\tilde{}\, -1 + \text{day} + (1 \mid \text{isol/r}), \text{data} = \text{insect2},$$
$$\text{family} = \text{binomial(link=cloglog))}.$$

The output from this model is summarized below:

```
> summary(ran.rep)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation)
['glmerMod']
Family: binomial  ( cloglog )
Formula: cbind(y, n_risk - y) ~ -1 + day + (1 | isol/r)
Data: insect2

AIC      BIC     logLik    deviance df.resid
3588.5   3635.3  -1784.2   3568.5      790

Scaled residuals:
Min      1Q     Median      3Q     Max
-4.1082 -1.0687 -0.5185  0.5864  9.8307

Random effects:
Groups Name        Variance Std.Dev.
r:isol (Intercept) 0.6489   0.8055
isol   (Intercept) 0.2650   0.5148
Number of obs: 800, groups:  r:isol, 100; isol, 20

Fixed effects:
      Estimate Std. Error z value Pr(>|z|)
day1  -4.2878     0.1922 -22.312  < 2e-16 ***
day2  -4.0506     0.1841 -22.000  < 2e-16 ***
day3  -3.1129     0.1611 -19.322  < 2e-16 ***
day4  -1.5541     0.1473 -10.548  < 2e-16 ***
day5  -1.2295     0.1479  -8.313  < 2e-16 ***
day6  -1.1827     0.1509  -7.838 4.58e-15 ***
day7  -1.8221     0.1645 -11.076  < 2e-16 ***
day8  -2.2696     0.1802 -12.596  < 2e-16 ***
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Notice that here the effect of replicates and replicates within isolates are both included in the model as random intercepts. The above model display shows that isolate-to-isolate variability (with estimated variance 0.265) has a smaller contribution than replicate within isolate variability (with estimated variance 0.649) in describing the response variability, however, both components are still quite appreciable.

It is noteworthy to mention that removing the term `(1 | isol:r)` reduces this model to model `ran.isol`. This change reduces the estimated variance of the grouping factor `isol (Intercept)`, but the change is tiny. However the change in the deviance and AIC is large, therefore this large amount of variation between replicates cannot be ignored and has to be considered in the model as its contribution is substantial to the overall fit.

The baseline hazard function estimate and cumulative hazard are plotted in Figures 4.11 and 4.12. They have similar behaviour to that explained for Figure 4.9.

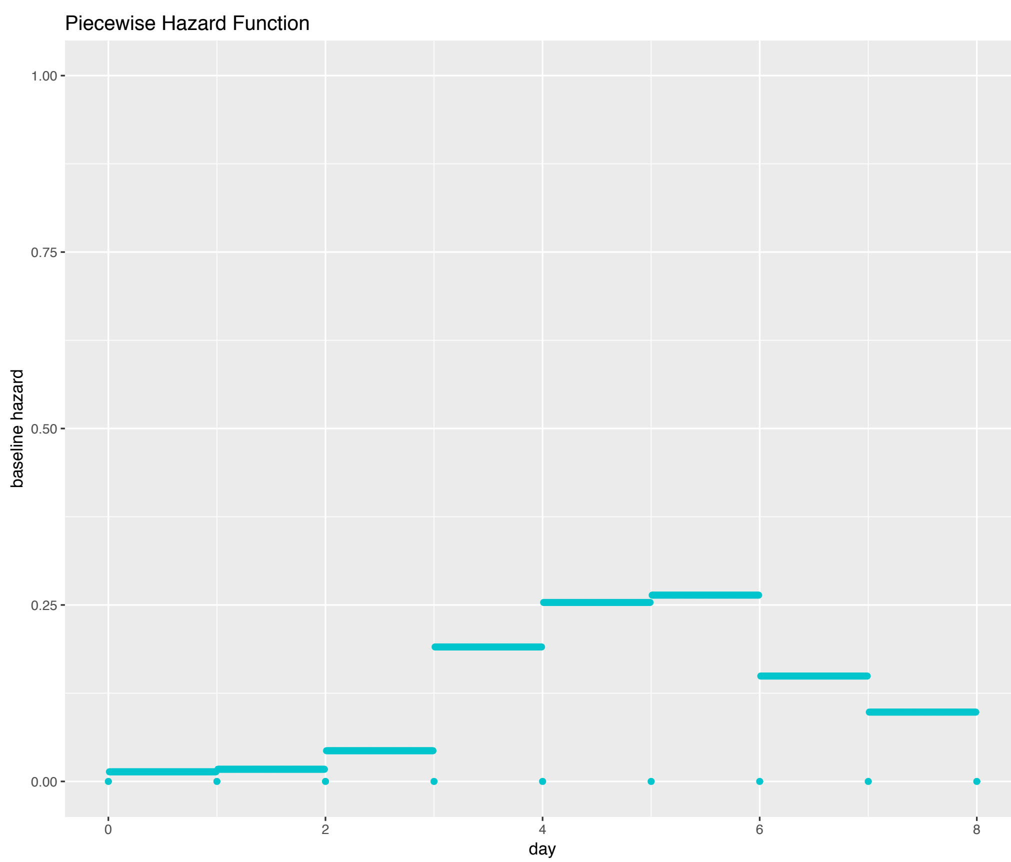


Figure 4.11: Piecewise Hazard Function for the GLMM (4.17)

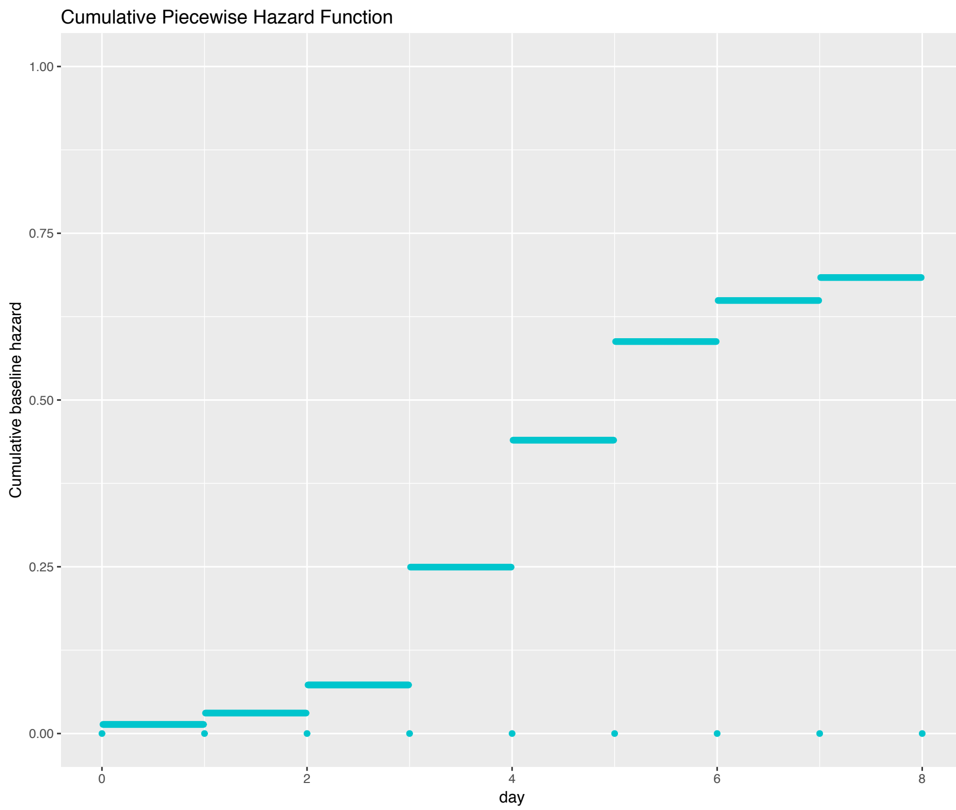


Figure 4.12: Cumulative Piecewise Hazard Function for the GLMM (4.17)

For the above fitted GLMM, we plot the estimated isolates random effects and group the similar values in Figure 4.13; this allows a comparison with the clusters formed in Figure 4.3 according to model `m1`. This figure suggests four possible clusters. Notice that according to Figure 4.3 the same number of clusters were suggested, nonetheless, one cluster contained only one isolate, 885, and was recognized by the model as an outlier. However, the model `ran.rep` describes the variations in such a way that fits the data well as we will see below.

In order to explore the random effects we consider their conditional modes and these can be obtained from the model (4.17) and are given below for both the replicates within each isolate (`r:isol`) and the isolates (`isol`). Notice that they are given on the linear predictor scale, i.e., `cloglog` scale for the mean..

```
R> ranef(ran.rep)
$‘r:isol’
      (Intercept)
1:282 -0.1157507044
1:309 -1.3249463190
```

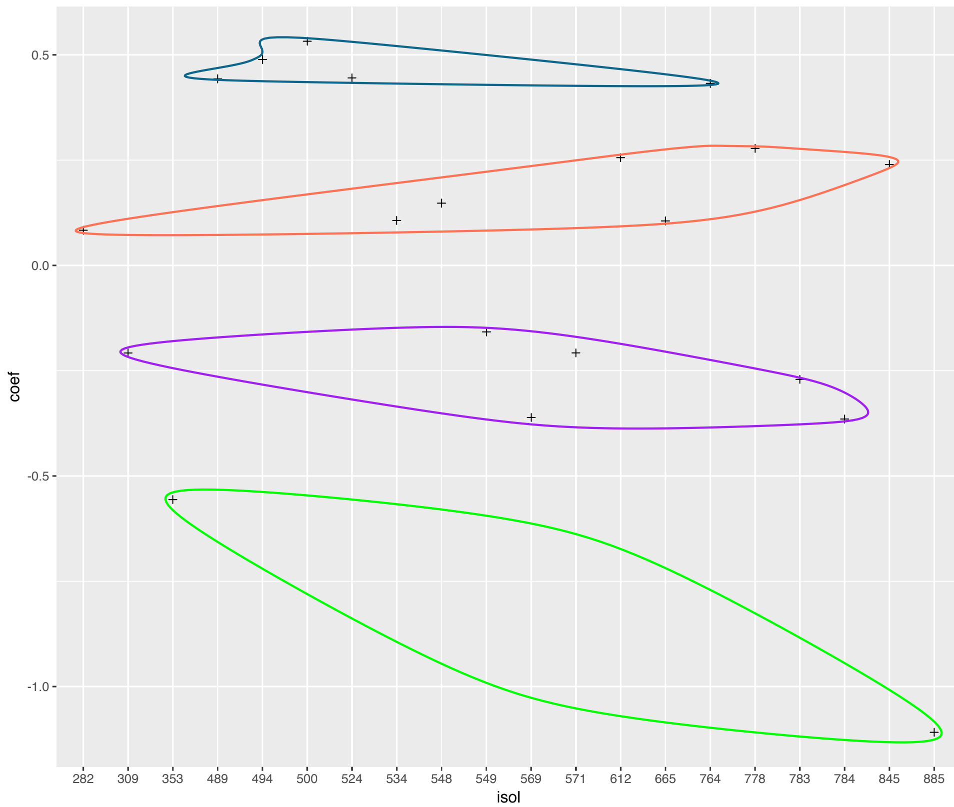


Figure 4.13: Clusters of isolates according to the random effects' estimated coefficients for the GLMM (4.17)

```
1:353 -0.4219174403
1:489 -1.4102353066
...
1:778 -1.1906786529
1:783 -0.2164693984
1:784 -0.3546036094
1:845  0.0958565958
1:885  0.3386020143
...
5:282  0.2866431606
5:309  0.6248908476
5:353  0.1275259737
...
5:783  0.7823222642
5:784  0.7177413871
```

```
5:845  0.1521122270
5:885 -1.5575172012

$isol
     (Intercept)
282  0.08349591
309 -0.20791248
353 -0.55604531
489  0.44298816
494  0.48891900
...
778  0.27780314
783 -0.27059294
784 -0.36475291
845  0.23950095
885 -1.10865583
```

These conditional models of the random effects are estimated by minimizing the mean squared error of prediction,

$$E(\tilde{u}-u)^2 = \int\int (\tilde{u}-u)^2 f(u,\boldsymbol{y})\ d\boldsymbol{y}\ du, \tag{4.18}$$

where u is considered as a typical scalar random effect, $f(u,\boldsymbol{y})$ is the joint density of $\boldsymbol{y}$ and u, and $\tilde{u}$ is the predictor of u. The random effect estimates/conditional modes obtained using this criterion (4.18) are referred to as the *Best Linear Unbiased Predictor* (BLUP), in the sense that random effects are basically random variables and unbiasedness for them is defined as to have the expected value of the predictor minus the random effect variable as zero, i.e. $E(\tilde{u}-u) = 0$ or $E(\tilde{u}) = E(u)$. It is also called a *shrinkage estimator*, meaning that $var(\tilde{u}) \leq var(u)$. That is BLUP estimates have lower variance/mean squared error than would estimates based on assuming the random effects were fixed effects. For more details on these, see (McCulloch and Searle, 2008, chapters 6, 13).

To illustrate how the BLUP estimates can be obtained, we consider obtaining the prediction for, say, the isolate 885, which is the first isolate in the dataset, on day 1. Notice that the sum of random effects predictions for the isolate 885 and for the first replicate within this isolate is equal to -0.7700538 ($= -1.10865583 + 0.3386020143$). The BLUP is obtained by combining this with the fixed effect estimate for the first day which is calculated as -4.287751. Therefore, the BLUP for the first replicate of isolate 885 on the first day is equal to the sum of fixed and random effects, i.e. $-4.287751 - 0.7700538 = -5.057805$. This can also be found by calling the R function `predict()` as below.

```
R> predict(ran.rep)[1]
-5.057805
```

Generally speaking, the BLUP estimate for y_{ijk} is the sum of k-th fixed and random effect estimated values in the j-th replicate within the i-th isolate.

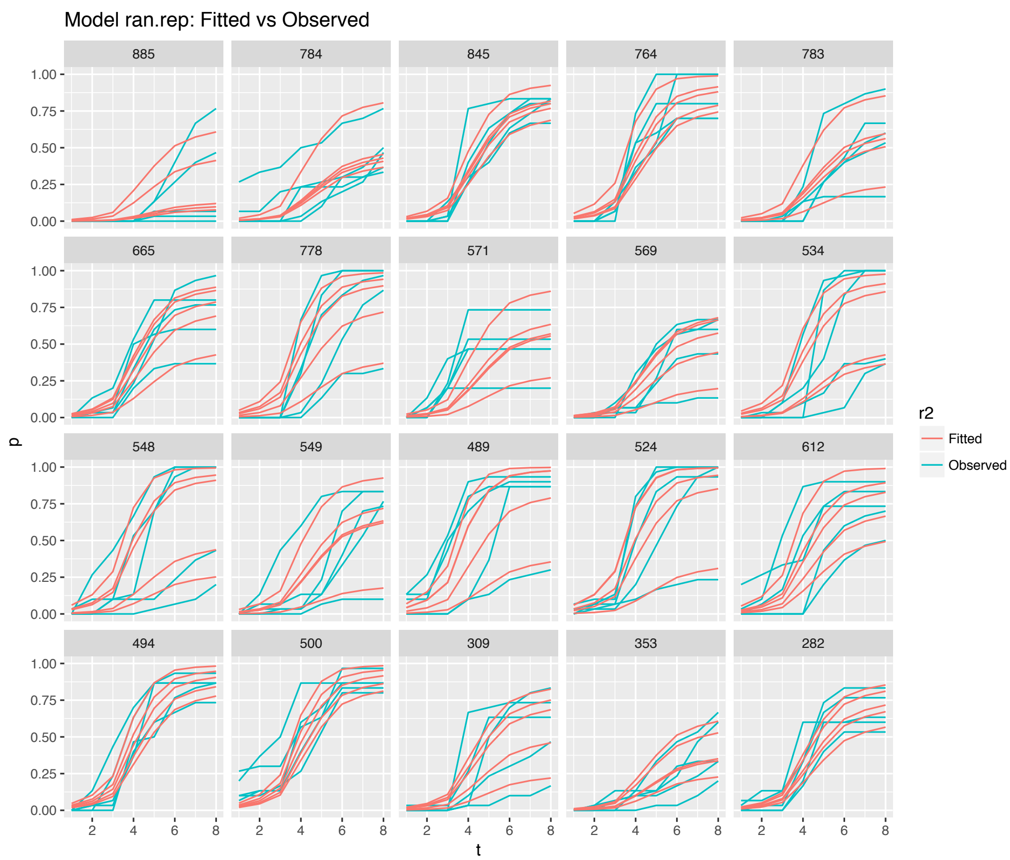


Figure 4.14: Cumulative Mortality for model (4.17)

Taking these BLUP estimates we can form predicted cumulative profiles and Figure 4.14 shows the plot of observed and predicted longitudinal profiles for the mixed effect model (4.17). Based on this plot we see that the model does quite well at capturing the observed patterns for most of the isolates.

The 95% confidence intervals for the fixed effects and the 95% prediction intervals accounting for fixed effects' uncertainty and isolate random effects' variance for model (4.17) are shown in Figure 4.15. Notice that they are in two panels with the same scale for the vertical axes for the sake of easy comparison.

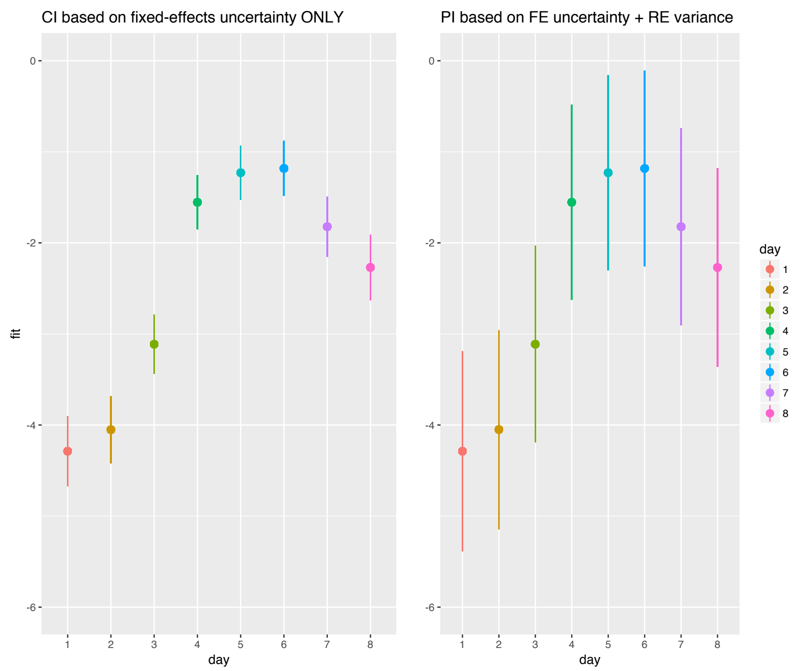


Figure 4.15: Left panel shows the confidence intervals for model (4.17) considering the fixed effects only and the right panel displays the prediction intervals obtained by considering the mixed effects. Notice that the predicted values are on the linear predictor scale, i.e., cloglog.

In order to test the hypothesis $H_0 : \sigma_{\texttt{isol}} = 0$ vs $H_a : \sigma_{\texttt{isol}} > 0$, we look at Figures 4.16 showing the prediction intervals for the random effect `isol`. Zero is within none of the prediction intervals for this factor which means that $\sigma_{\texttt{isol}}$ is significant and that we should reject H_0 in favour of H_a.

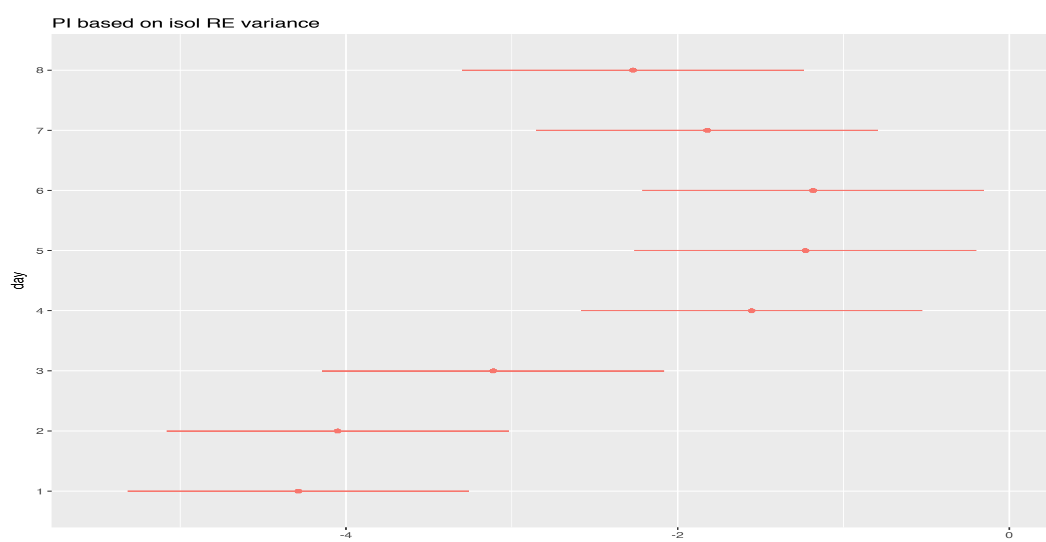


Figure 4.16: Prediction intervals for the isolate random effect based on model (4.17). Predicted values are on cloglog scale.

4.3.4 Comparison between models

Here we compare the models fitted above using goodness of fit in terms of their residual deviances and BIC (Bayesian Information Criterion), although the AIC (Akaike Information Criterion) criterion could also be used and is almost always equivalent in terms of conclusions. The smaller the residual deviance and the BIC, the better fit the model is. The information criteria are defined as

$$\begin{aligned}\text{AIC} &= -2\,\text{loglik} + 2p \\ \text{BIC} &= -2\ \text{loglik} + p\log(n),\end{aligned}$$

where p is the number of free parameter estimates in the model, n is the number of observations and loglik represents the maximized log-likelihood value.

The first two fixed effects models `m1` and `m2` have the same fixed effect terms, but different link functions, so they are directly comparable using the `anova` function in `R`. Comparison of models with different fixed effects within the mixed model framework is not valid when REML is used. The reason is that in the REML method, the likelihood of linear combination of the terms not involving the fixed effects is maximized. So with different fixed effects in the models, the linear combinations are different and so the obtained restricted likelihood estimates based on them are not comparable, see (Faraway, 2006, page 163). The residual deviance for the model `m1` is lower by about 236 units and so it is preferable. Notice that here the BIC difference between models `m1` and `m2` is equal to the residual deviance as they have equal degrees of freedom.

```
R> anova(m1,m2)
Analysis of Deviance Table

Model 1: cbind(y, n_risk - y) ~ -1 + day + isol
Model 2: cbind(y, n_risk - y) ~ -1 + day + isol
   Resid. Df Resid. Dev Df Deviance
1        751     2948.3
2        751     3184.9  0  -236.52

R> BIC(m1,m2)
   df      BIC
m1 27 4245.105
m2 27 4481.627
```

For the pairwise comparison of model `m1`, and the mixed effects models `ran.rep` and `ran.isol`, we should first consider the design of the experiment that the data came from. The sampling plan for the data as mentioned in Section 4.2 is such that the effect of isolates can be considered as random, to say something about the general population of isolates rather than the specific set of isolates that happen to have been applied. This suggests that a mixed effect model can capture this variability better than a fixed effects model. Furthermore, the variation between replicates varies from one isolate to the another showing that this variation should be included in the model to avoid the overdispersion, suggesting that the two random effects term model fits better.

In order to compare a fixed and a mixed effect model we can compare their BIC criteria. This is because of the fact that the deviances presented by `glm` and `glmer` are not on the same scale, with the first one reporting the $-2\times$log-likelihood plus $2\times$log-likelihood of the saturated model and the second one reporting $-2\times$log-likelihood.

According to the output below, the BIC of mixed effect models are both smaller than that of the fixed model `m1` and this shows that the mixed effect models have a better fit .

```
R> BIC(m1,ran.rep)
        df      BIC
m1      27 4245.105
ran.rep 10 3635.328
```

```
R> BIC(m1, ran.isol)
         df      BIC
m1       27 4245.105
ran.isol  9 4209.552
```

The mixed effect models can be compared using the BIC criterion and also the difference can be tested using the `anova` function, as follows.

```
R> > BIC(ran.isol,ran.rep)
         df      BIC
ran.isol  9 4209.552
ran.rep  10 3635.328

R> anova(ran.isol,ran.rep)
Data: insect2
Models:
ran.isol: cbind(y, n_risk - y) ~ -1 + day + (1 | isol)
ran.rep: cbind(y, n_risk - y) ~ -1 + day + (1 | isol/r)
         Df    AIC    BIC  logLik deviance  Chisq Chi  Df Pr(>Chisq)
ran.isol  9 4167.4 4209.6 -2074.7   4149.4
ran.rep  10 3588.5 3635.3 -1784.2   3568.5  580.91      1  < 2.2e-16 ***
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

One can see that model `ran.rep` is significantly better than `ran.isol` according to the Chi-square test as well, as the BIC criterion. According to the highly significant p-value in the likelihood ratio test, by the `anova` function, based on the model `ran.rep` and its reduced version `ran.isol`, `(1 | isol)` is removed, we conclude that H_0 in testing $H_0 : \sigma_{\texttt{r:isol}} = 0 \quad \text{vs} \quad H_a : \sigma_{\texttt{r:isol}} > 0$, is rejected.

4.4 GLMM with a Non-parametric Distribution over Random Effects

As mentioned in the introduction above, a choice of an appropriate random effect distribution has been studied before by authors, however, when there is not enough prior knowledge to decide what distribution might fit the random effects best, using a non-parametric distribution would seem a reasonable approach to avoid the model misspecification. Aitkin (1996) used the idea of a non-parametric random effect distribution, as suggested by Laird (1978), and assumed its density to be an unknown mixture distribution. Einbeck and Hinde (2006) followed this approach for fitting a GLMM with non-parametric random effect distribution. We use Einbeck and Hinde (2006)'s method in this section as to do so also leads to an induced clustering of the isolates. This will be discussed in a little more detail later.

The summary of Einbeck and Hinde (2006) model fitting approach is given as follows:

1. Suppose $y_1, \ldots, y_n$ are a set of observations from an exponential family distribution, $f(y_i|\beta, \phi)$, with dispersion parameter ϕ.

2. Let $x_1, \ldots, x_n$ be a set of explanatory variables.

3. The linear predictor η relates the mean response to the predictors through a link function $h(\cdot)$, as usual in the GLM, however, to account for additional unexplained variability of the individual observations and overcome the problem of overdispersion, a random effect term z_k with mixture density $g(\cdot)$ is added to the linear predictor, as follows
$$\eta_i = x_i'\beta + z_{ki}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, K.$$
$\boldsymbol{z_k} = (z_{k1}, \ldots, z_{kn})$ is assumed to be a K level factor, with K unknown.

4. For a specific assumed value of K, the likelihood for the observations can be approximated by a finite mixture as below:

$$L = \prod_{i=1}^{n} \int f\left(y_i|z_{ki}, \beta, \phi_{ki}\right) g(z_{ki})\, dz_{ki} \approx \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} f_{ik}\pi_{ki} \right\}, \tag{4.19}$$

   where $f_{ik} = f(y_i|z_k, \beta, \phi)$, z_k are the mass points and π_k are their masses.

5. The partial derivatives of the log-likelihood, l, score equations for the unknown z_k, β and ϕ turn out to be weighted versions of a single distribution score equations with weights $w_{ik} = \frac{\pi_k f_{ik}}{\sum_{l=1} \pi_l f_{il}}$. These weights can be interpreted as posterior probabilities for observation i coming from component k of the mixing distribution for $\boldsymbol{z}$. Then according to the fact that $\sum_k \pi_k = 1$, the score equation for the mixing proportions is adjusted as

$$\frac{\partial l - \lambda\left(\sum \pi_k - 1\right)}{\partial \pi_k} = 0,$$

   which gives the maximum likelihood estimate of the mixing proportion $\hat{\pi}_k = \frac{1}{n}\sum_i w_{ik}$, i.e., the average posterior probability of the observations for belonging to component k.

6. The unknown parameters z_k, β and ϕ are estimated through an EM algorithm, where in the E-step the mixture component membership w_{ik} is estimated and in the M-step unknown parameters are estimated by fitting GLM to each levels of factor z_k with weights w_{ik} forming a weighted GLM fitting. This procedure is repeated for different values of K, typically increasing K until there is no substantial change in the maximized likelihood value.

Remark. *We use the* `R` *package* `npmlreg` *for computations which includes all the steps explained above.*

4.4.1 Model

Here we fit a mixed effect model to the binary data from Section 4.2 with considering the factor `day` as a fixed effect and the effects of isolates as random. Furthermore, we do not specify the usual normal distribution for the random effects, but instead we consider it as a non-parametric K-component mixture distribution. The model formulation in `R` is given by

```
npRE<-allvc(cbind(y, n_risk - y) ~ day, random = ~ 1 | isol, data=insect2, k=4,
                 family = binomial(link=cloglog), random.distribution = "np").
```

A summary of the fitted model output is given below.

```
> summary(npRE)

Call:  allvc(formula = cbind(y, n_risk - y) ~ day, random = ~1 | isol,
              family = binomial(link = cloglog), data = insect2, k = 4,
              random.distribution = "np")

Coefficients:
       Estimate  Std. Error  t value
day2   0.226284  0.1755645   1.288894
day3   1.140009  0.1509433   7.552562
day4   2.611628  0.1362445  19.168687
day5   2.792190  0.1374511  20.314060
day6   2.716397  0.1410479  19.258679
day7   2.010426  0.1563545  12.858128
day8   1.541507  0.1732996   8.895040
MASS1 -5.072465  0.1634350 -31.036590
MASS2 -4.386305  0.1369329 -32.032514
MASS3 -3.831384  0.1332326 -28.757115
MASS4 -3.444399  0.1333980 -25.820473

Mixture proportions:
MASS1       MASS2       MASS3       MASS4
0.09888987  0.30105550  0.35109148  0.24896315

Random effect distribution - standard deviation:    0.4948603

-2 log L:     4143     Convergence at iteration  8
```

Increasing the number of mass points K from 4 to 5, does not make a major difference between the estimates of the 4th and 5th mass points and nor between the model disparities ($-2\log L$). Therefore we keep K equal to 4 to have four distinct mass points.

Now looking at the structure of the EM trajectories plot in Figure 4.17, one can see that the model converges rapidly and in the lower plot we have a suggestion of how the individual data points may be allocated to the four clusters.

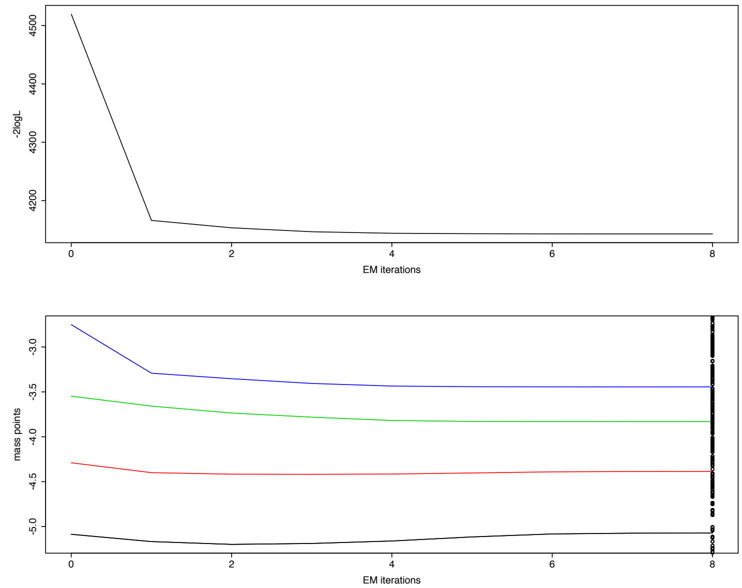


Figure 4.17: Plot of disparity trend at the top and that of EM trajectories at the bottom

In order to get an idea about how similar, or different, the two mixed effects models are, with parametric and non-parametric random effects distributions, we look at the structure of the predicted random effects for each isolate. Figures 4.18 pictures the predicted random effects from the non-parametric random effect model and model with random effects for both isolate and replicate against each other. One can see that they have generally similar pattern but are not quite the same.

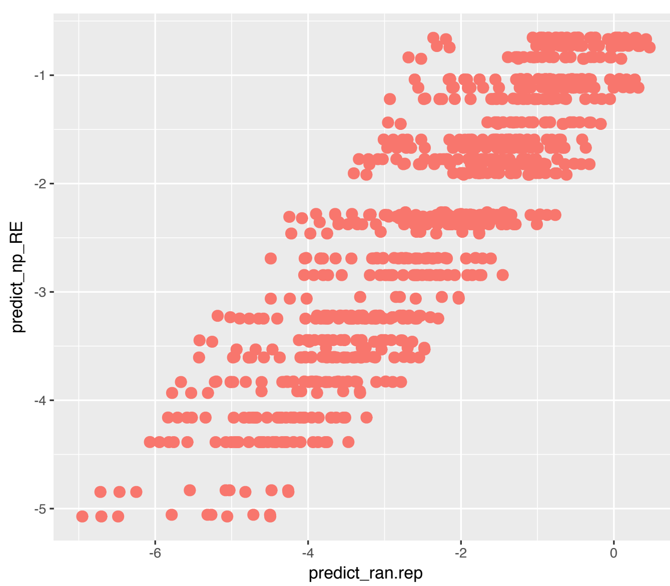


Figure 4.18: Plot of predictions given by `npRE` versus those given by `ran.rep`

In order to find out which isolates are allocated to which cluster, we use the posterior probability weights. As the posterior probabilities indicate the probability of observations belonging to the classes/clusters so by finding the mean of them over classes we can assign isolates to the class with largest mean value of posterior probability. Although looking at the observation level, in practice there might be a few outlier observations assigned to a different class than what was suggested by the mean value, however, we assume that there is no outlier in the posterior weights that could heavily affect the mean and therefore we restrict ourselves to the overall isolate level. Figure 4.19 demonstrate how the isolates are allocated to the clusters based on the fitted non-parametric model.

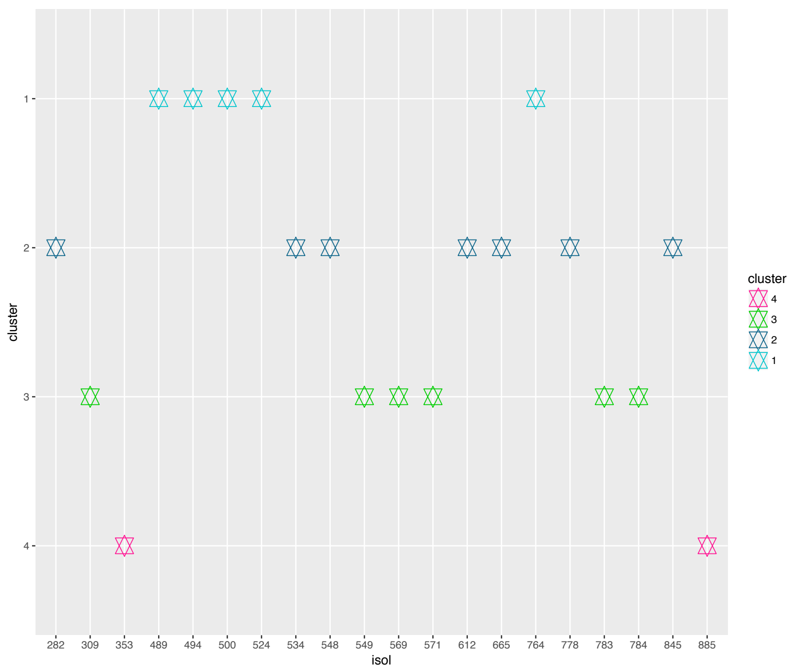


Figure 4.19: Clusters of isolates based on the non-parametric model

The grouping of isolates is almost everywhere in line with those from the two other models: (1) the model with one random effect term, see Figure 4.10 and (2) the model with fixed effects only with `cloglog` link function, see Figure 4.3. Except that here there is no outlier isolate whereas isolate number 885 was described as different from all other isolates by the other two models. Moreover, the grouping by this non-parametric model is exactly the same as in Figure 4.13 categorized by model `ran.rep`.

Hence, this shows that the presence of the second random effect term discussed in Sections 4.3.3 and 4.3.4 still seems crucial. Table 4.1 shows the number of isolates within each cluster formed by these four models.

Table 4.1: A comparison of number of isolates within clusters

Model	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Fixed effects only	5	7	7	1
One random term	5	7	7	1
Non-parametric	5	7	6	2
Two random terms	5	7	6	2

4.5 Models with the time trend

As can be seen in the previous sections we have considered different fixed and mixed effect models with one and two random effect terms so far. In this section, we account for additional possible time trend in the model. This means that we consider a time covariate as the time duration that insects are exposed to toxicity. This variable is denoted by `time`. Apart from the factor covariate `day` which has a constant effect on the response over each day, the variable `time` adds the linear effect of time on mortality. Here, we consider adding this variable to a fixed and a mixed effects model. The dataset used in this section is called `insect3` in which the `time` variable is included. The data structure is given by

```
R>  str(insect3)
'data.frame': 800 obs. of  11 variables:
$ isol   : Factor w/ 20 levels "282","309","353",..: 20 20 20 20 20 20 20 20 20 20 ...
$ r      : int  1 1 1 1 1 1 1 1 2 2 ...
$ y      : int  0 0 0 0 4 4 4 2 0 0 ...
$ p      : num  0 0 0 0 0.133 ...
$ n      : int  30 30 30 30 30 30 30 30 30 30 ...
$ day    : Factor w/ 8 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 1 2 ...
$ time   : num  1 2 3 4 5 6 7 8 1 2 ...
```

4.5.1 Fixed effect model

As mentioned above, here we add a time trend to the fixed effect model (4.14). Basically we add the linear time trend within each isolate, treating `time` as a fixed effect slope, additional to the effect of isolates themselves. This is done by adding the interaction term `isol*time`, noting that the main effect of `time` is completely aliased with the factor `day` that estimates the baseline hazard. The model formulation in `R` is given below followed by the model summary.

```
R> m0.fix <- glm(cbind(y, n_risk - y) ~ -1 + day + isol*time,
                    family = binomial(link = cloglog), data = insect3)
```

```
R> summary(m0.fix)

Call:
glm(formula = cbind(y, n_risk - y) ~ -1 + day + isol * time,
family = binomial(link = cloglog), data = insect3)

Deviance Residuals:
Min       1Q   Median       3Q      Max
-4.7442  -1.3585  -0.4947   0.7111   6.7962

Coefficients: (1 not defined because of singularities)
             Estimate   Std. Error z value Pr(>|z|)
day1         -3.86103    0.29977 -12.880  < 2e-16 ***
day2         -3.52782    0.23246 -15.176  < 2e-16 ***
day3         -2.58682    0.16052 -16.116  < 2e-16 ***
day4         -1.16533    0.10908 -10.683  < 2e-16 ***
day5         -1.12212    0.11292  -9.937  < 2e-16 ***
day6         -1.41826    0.15840  -8.954  < 2e-16 ***
day7         -2.41211    0.22844 -10.559  < 2e-16 ***
day8         -3.21315    0.30575 -10.509  < 2e-16 ***
isol309      -1.48686    0.54121  -2.747  0.00601 **
isol353      -4.34836    0.69146  -6.289 3.20e-10 ***
isol489       1.32594    0.43892   3.021  0.00252 **
isol494       0.14805    0.45854   0.323  0.74679
isol500       1.39915    0.43586   3.210  0.00133 **
isol524      -0.17700    0.46830  -0.378  0.70546
isol534      -2.25464    0.52800  -4.270 1.95e-05 ***
isol548      -0.66649    0.49719  -1.341  0.18008
isol549      -2.66728    0.56587  -4.714 2.43e-06 ***
isol569      -1.68589    0.57175  -2.949  0.00319 **
isol571       1.92320    0.45316   4.244 2.20e-05 ***
isol612       0.47132    0.46658   1.010  0.31242
isol665      -0.23811    0.48585  -0.490  0.62407
isol764      -0.26542    0.46886  -0.566  0.57133
isol778      -2.16174    0.50892  -4.248 2.16e-05 ***
isol783      -2.97787    0.58255  -5.112 3.19e-07 ***
isol784      -1.33005    0.56775  -2.343  0.01915 *
isol845      -1.09811    0.49100  -2.236  0.02532 *
```

```
isol885       -6.01004    0.88607  -6.783 1.18e-11 ***
time                NA         NA      NA       NA
isol309:time  0.23339    0.10933   2.135  0.03279 *
isol353:time  0.65454    0.12380   5.287 1.24e-07 ***
isol489:time -0.22801    0.09849  -2.315  0.02060 *
isol494:time  0.08466    0.09977   0.849  0.39610
isol500:time -0.20020    0.09803  -2.042  0.04113 *
isol524:time  0.12857    0.10047   1.280  0.20063
isol534:time  0.46356    0.10509   4.411 1.03e-05 ***
isol548:time  0.15171    0.10343   1.467  0.14245
isol549:time  0.47611    0.10912   4.363 1.28e-05 ***
isol569:time  0.23215    0.11418   2.033  0.04203 *
isol571:time -0.55966    0.10835  -5.165 2.40e-07 ***
isol612:time -0.07163    0.10142  -0.706  0.48005
isol665:time  0.06314    0.10341   0.611  0.54146
isol764:time  0.15948    0.10078   1.582  0.11354
isol778:time  0.49493    0.10264   4.822 1.42e-06 ***
isol783:time  0.50189    0.11158   4.498 6.85e-06 ***
isol784:time  0.15063    0.11374   1.324  0.18541
isol845:time  0.27979    0.10167   2.752  0.00593 **
isol885:time  0.83095    0.14845   5.598 2.17e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 25271.7  on 778  degrees of freedom
Residual deviance:  2671.2  on 732  degrees of freedom
AIC: 3879.5

Number of Fisher Scoring iterations: 6
```

Slope estimates associated with isolates 353 and 885 are much larger than others. This is consistent with other models. For instance, in the non-parametric model in the previous section we see that the aforementioned isolates are classified together in mass point four.

The following figure shows how the values of slope with respect to `time` and mid-response estimates for isolates vary, where mid-response is defined as (Intercept + slope$*$4.5), and 4.5 is the midpoint/mean value of the `time` variable. Looking at the slope as an isolate-specific measure of speed of death, we see that isolates 353 and 885 have a very high slope but low mid-point resposnse, suggesting that they are rather slow acting compared to some other more effective isolates.

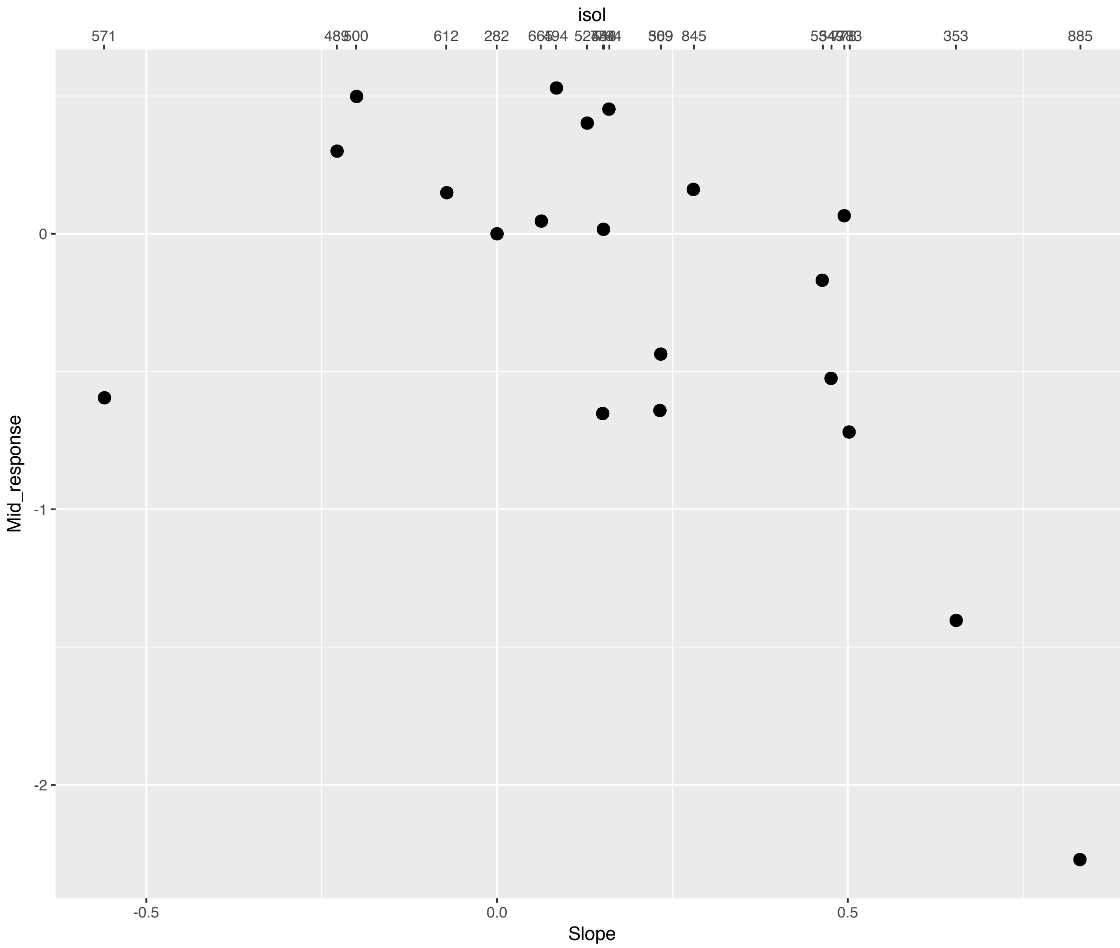


Figure 4.20: Plot of intercept and slope estimates for isolates

Now we compare the models `m0.fix`, `m1` and `ran.rep` using the BIC criterion as given below. One can conclude that `m0.fix` has a better fit to the data than the fixed effect model `m1` in terms of a lower BIC, however, it is still not as good as the mixed effects model `ran.rep`.

```
R> BIC(m1, m0.fix)
       df      BIC
m1     27 4245.105
m0.fix 46 4095.002
```

```
R> BIC(ran.rep, m0.fix)
        df      BIC
ran.rep 10 3635.328
m0.fix  46 4095.002
```

4.5.2 Mixed effect model

Time-varying isolate random effect

Here we consider a mixed effect model where `isol` is included as a random intercept and the random slope with respect to variable `time` is included through a scaled version of `time`, by dividing the difference of this variable from its mean by its standard deviation, and uncorrelated to the random intercept. Scaling the `time` variable is done to avoid failure of convergence (computational problems) and also to remove correlation with random factor `isol` (and possible confounding of the variance estimates). Furthermore as the effect of time is nested within isolates, therefore the desired model is formulated and summarized as below. Notice that the additive term 0 in the formula indicates the removed correlation explained above.

```
R> m1.ran<-glmer(cbind(y, n_risk - y)  ~  -1 + day + (1 | isol) +
                    (0 + scale(time) | isol), data = insect3,
                    family = binomial(link=cloglog))

R> summary(m1.ran)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation)
['glmerMod']
Family: binomial  ( cloglog )
Formula: cbind(y, n_risk - y) ~ -1 + day + (1 | isol) + (0 + scale(time) | isol)
Data: insect3

AIC      BIC     logLik    deviance df.resid
3977.3   4024.2  -1978.7   3957.3      790

Scaled residuals:
Min      1Q     Median      3Q     Max
-3.6533 -1.0366 -0.4235  0.7819 17.1101

Random effects:
Groups Name        Variance Std.Dev.
isol   (Intercept) 0.3730   0.6107
isol.1 scale(time) 0.4266   0.6532
Number of obs: 800, groups:  isol, 20

Fixed effects:
```

```
      Estimate Std. Error z value Pr(>|z|)
day1  -4.6568     0.3030 -15.368  < 2e-16 ***
day2  -4.1675     0.2458 -16.958  < 2e-16 ***
day3  -3.0628     0.1858 -16.487  < 2e-16 ***
day4  -1.4720     0.1469 -10.022  < 2e-16 ***
day5  -1.2545     0.1472  -8.523  < 2e-16 ***
day6  -1.3660     0.1755  -7.782 7.12e-15 ***
day7  -2.1613     0.2278  -9.490  < 2e-16 ***
day8  -2.7492     0.2880  -9.546  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Now we compare the BIC criterion of this model with the single random effect model `ran.isol` from Section 4.3.3 and the best model so far which is the two random effect model `ran.rep`. As we expected when entering the time trend in the model, the model has a better fit with lower BIC, although the model `ran.rep` still has a better fit.

```
R> BIC(m1.ran, ran.isol)
df      BIC
m1.ran   10 4024.160
ran.isol  9 4209.552
```

```
R> BIC(m1.ran, ran.rep)
df      BIC
m1.ran  10 4024.160
ran.rep 10 3635.328
```

Time-varying isolate and replicate random effects

Here we consider a more complicated model with both a time-varying random intercept and slope within isolates and a random effect for replicates within isolates. This is denoted by `( scale(time) | isol/r )` which expands to `( scale(time) | isol) + ( scale(time) | isol:r )`. The model summary is given below.

```
R> summary(ran.rep.t)
Generalized linear mixed model fit by maximum likelihood
(Laplace Approximation) ['glmerMod']
Family: binomial  ( cloglog )
Formula: cbind(y, n_risk - y) ~ -1 + day + ( scale(time) | isol/r)
Data: insect3

AIC      BIC      logLik deviance df.resid
3041.7   3107.2  -1506.8   3013.7      786
```

```
Scaled residuals:
Min      1Q     Median      3Q     Max
-3.1594 -0.7557 -0.2368  0.4232 18.1357

Random effects:
Groups Name        Variance Std.Dev. Corr
r:isol (Intercept) 0.7599   0.8717
scale(time)        1.2141   1.1019   -0.17
isol   (Intercept) 0.4128   0.6425
scale(time)        0.6235   0.7896   -0.37
Number of obs: 800, groups:  r:isol, 100; isol, 20

Fixed effects:
      Estimate  Std. Error z value Pr(>|z|)
day1  -6.0291     0.4495 -13.412  < 2e-16 ***
day2  -5.0651     0.3572 -14.181  < 2e-16 ***
day3  -3.5612     0.2659 -13.393  < 2e-16 ***
day4  -1.6623     0.1944  -8.549  < 2e-16 ***
day5  -1.2974     0.1670  -7.766 8.07e-15 ***
day6  -1.3262     0.1903  -6.968 3.21e-12 ***
day7  -2.1236     0.2575  -8.248  < 2e-16 ***
day8  -2.8404     0.3430  -8.280  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Correlation of Fixed Effects:
      day1   day2   day3   day4   day5   day6   day7
day2  0.894
day3  0.863  0.884
day4  0.738  0.793  0.883
day5  0.388  0.472  0.614  0.816
day6 -0.046  0.043  0.197  0.466  0.787
day7 -0.303 -0.227 -0.093  0.170  0.564  0.840
day8 -0.420 -0.356 -0.242  0.001  0.413  0.754  0.866
convergence code: 0
Model failed to converge with max|grad| = 0.122558 (tol = 0.001, component 1)
```

According to the above output, the correlation -0.17 indicates the estimated random slope within isolates and the correlation -0.37 indicates that between isolates. Although the model has not converged, it only seems to be due to insufficient data for estimation, see Bolker et al. (2017). We considered removing the correlation between the intercept and slope within the grouping factors, `(0 + scale(time) | isol) + ( 0 + scale(time) | isol:r )`, but it did not help with convergence. However, the deviance value of 3013.7 is the lowest value among all other previous models and suggests an improved fit and note that the intercept and slope random effects within `isol` and `r:isol` are not highly correlated. To explore this model in the sense of goodness of fit, we look at the plot of fitted predicted profiles compared to the observed in Figure 4.21, as we have done for previous models.

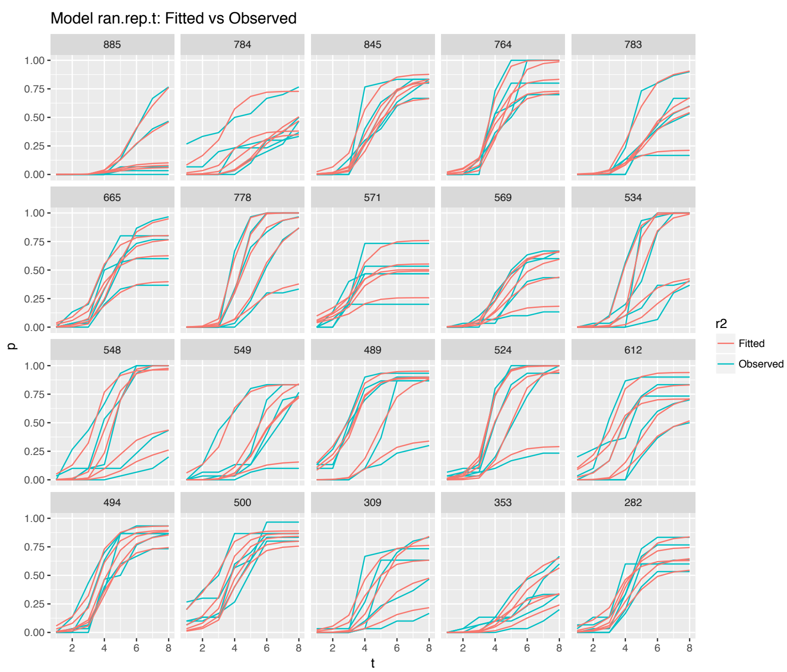


Figure 4.21: Plot of fitted and observed values based on model `ran.rep.t`

One can see from the plot that the observed values are predicted very well. The variation between the replicates is captured by the model quite well and there does not seem to be any problem with the model fitting. Also there is no strong evidence of overdispersion, or unexplained variation. Random effects for this model are considered at the isolate level and at the individual replicate level within isolates, as illustrated below.

```
$‘r:isol’
      (Intercept) scale(time)
1:282 -0.05878564  0.17487639
1:309 -1.46090234  0.52586779
1:353  0.02962399 -0.79329215
1:489 -1.28862307  0.63788769
...
5:783  0.93870597  0.25527156
```

```
5:784  0.96950881 -1.92465573
5:845  0.23880942 -0.05999537
5:885 -1.29185365 -0.60764355

$isol
    (Intercept) scale(time)
282  0.27846001 -0.53032119
309 -0.20649694  0.11410410
353 -0.84125536  0.71483874
489  0.45995898 -0.62105021
...
494  0.66470770 -0.43120675
784 -0.44588017 -0.07725509
845  0.36325607 -0.04014566
885 -1.42796865  0.60997231
```

Next, we consider the possible classification of the isolates into groups. This is shown in Figure 4.22.

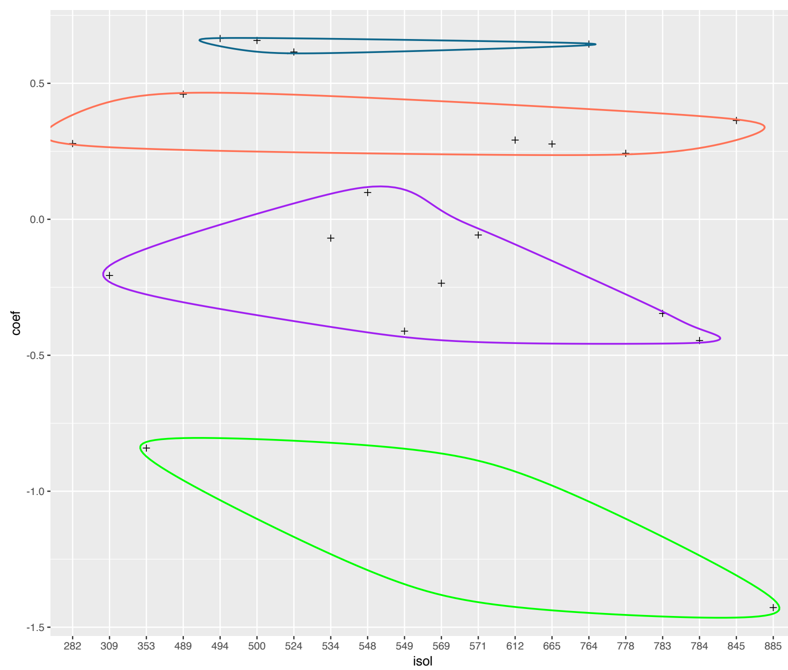


Figure 4.22: Clusters of isolates based on model `ran.rep.t`

This appears to be similar to the classification by `ran.rep` and again four clusters are formed. Although clusters are not exactly the same, they are quite similar. All in all, the model `ran.rep.t` seems to be better than `ran.rep` even though it has not fully converged.

4.5.3 Non-parametric random effect model

We now extend the non-parametric model to allow for a random intercept for `isol` and a random slope `time` between isolates using the following code. Notice that this model corresponds to the parametric model `m1.ran`.

```
npRE_tt <- allvc(cbind(y, n_risk - y)  ~  day, random = ~ time | isol,
```

```
                  data = insect3, k = 4, family = binomial(link = cloglog),
                  random.distribution = "np")
```

The summary of the model is given below.

```
> summary(npRE_tt)

Call:  allvc(formula = cbind(y, n_risk - y) ~ day, random = ~time | isol,
               family = binomial(link = cloglog), data = insect3, k = 4,
               random.distribution = "np")

Coefficients:
             Estimate    Std. Error      t value
day2       -0.002066877 0.17611630  -0.01173586
day3        0.619200940 0.15413632   4.01722939
day4        1.725577402 0.14955034  11.53843848
day5        1.470591934 0.16950742   8.67567901
day6        0.897757404 0.19852131   4.52222187
day7       -0.361445507 0.24230175  -1.49171645
day8       -1.408256236 0.29059454  -4.84612072
MASS1      -8.956415710 0.51992564 -17.22633986
MASS2      -6.061548950 0.26481812 -22.88947952
MASS3      -4.638347654 0.18642348 -24.88070564
MASS4      -2.637533849 0.14601027 -18.06402909
MASS1:time  0.990231867 0.09073900  10.91296865
MASS2:time  0.607032562 0.05659445  10.72600866
MASS3:time  0.460180332 0.04558615  10.09474015

Mixture proportions:
MASS1       MASS2       MASS3       MASS4
0.09999738  0.25107430  0.48708480  0.16184352

Random effect distribution - standard deviation:    1.674654

-2 log L:     4021.9     Convergence at iteration  9
```

A simple comparison between the above non-parametric model and the one without the `time` in shows that accounting for the time trend in capturing the variability is reasonable and leads to a more adequate model with a larger log-likelihood. To be more specific, the $-2\times$log-likelihood is smaller in the model with time trend on 3 degrees of freedom. Figures 4.23 demonstrates how the isolates have got allocated to four clusters based on the non-parametric model with the time trend in. In terms of comparison with the allocations in the non-parametric model in Section 4.4, cluster 4 is exactly the same in both and there are similarities in allocations within the other clusters. Figure 4.24 shows the EM trajectories and how the mass points are formed over EM algorithm iterations from the model with time trend. Comparing that with Figure 4.17, the clusters are more distinct. Also to compare this model with its parametric version `m1.ran`, we look at their log-likelihood values showing a superiority for model `m1.ran` with a higher value of the log-likelihood.

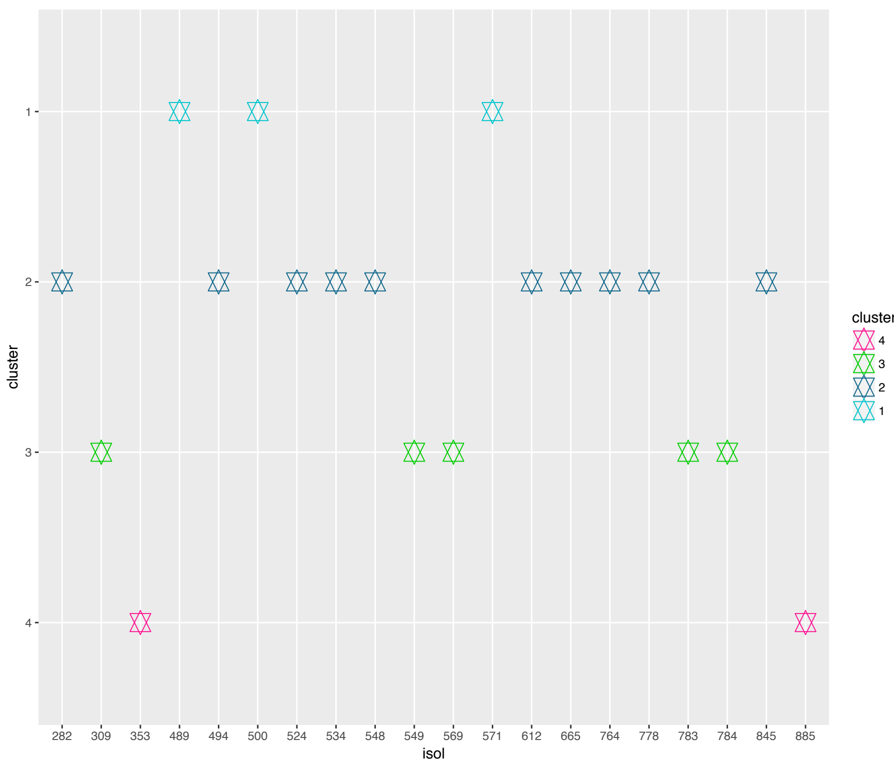


Figure 4.23: Clusters of isolates based on the non-parametric model with time trend

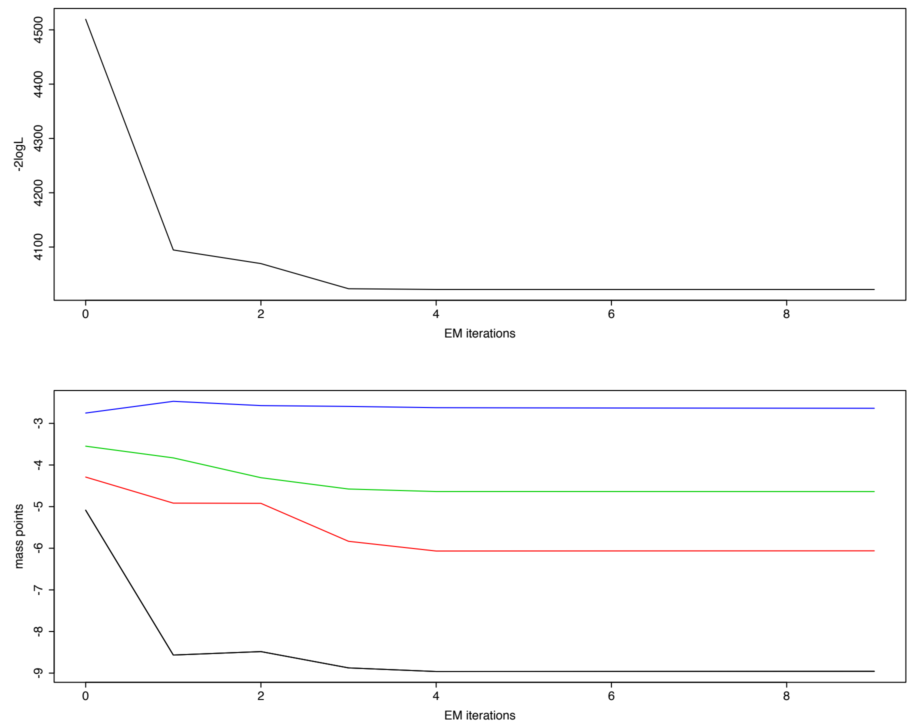


Figure 4.24: Clusters of isolates formed based on the non-paramtric model with time trend, `npRE_tt`.

4.6 Discussion

Mixed effect models have provided a useful tool for analyzing clustered data arising from experiment with grouped individuals, correlated outcome data, or grouped correlated data. Examples of such situations could be nested students within school, longitudinal data, or lifetime follow-up of patients in different hospitals over time, where the outcome data cannot be simply analyzed by a one- or two-way analysis of variance. The random effect models predict for random factor levels so that the results of modeling can be generalized to conclude the similar results if a new set of data are collected from the same population, whereas the fixed effect models could only compare the means of specific factor levels. Nowadays, since the random effect models are beneficial in analyzing lots of data structures, they have a widespread use.

Generalized linear mixed models are flexible tools including both fixed and random effects in the model. They can be fitted with software `R` with different packages, such as the most well-known packages `lme4` and `nlme`, although in this chapter we have only used the former. Notwithstanding the fact that GLMMs are very useful, they are computationally difficult and all of the issues that arise with regular linear or generalized-linear modeling can apply more severely to mixed models; for instance, inadequacy of p-values alone for thorough statistical analysis or the need to understand how models are parameterized. Moreover, failure to convergence problems can happen when the model is misspecified for example when a factor with a few levels is treated as random could cause imprecise estimates of random effects, or in a more severe case could make convergence problems, or estimate the variance of random effects as zero which is contrary with the assumption of being a random factor. Notice that the convergence problem could also happen due to insufficient data for estimation.

It is worth pointing out that sometimes, when having mostly (not perfectly) collinear predictors, the lack of convergence problems can be overcome with centering and scaling the predictors. This is what we have done in Section 4.5.2 with the `time` predictor which is both aliased to the factor `day` and correlated with the intercept, but scaling essentially removes this correlation between them and remedies the convergence problem.

Plots of the prediction intervals for the random effect are a helpful visual tool to test if the variance of a random effect is zero or not. In the other words, they are useful to examine if the effect of that factor is remarkable or not. However, there are situation, where zero is within most or all of the prediction intervals but one may believe that its effect should be important, according to some prior knowledge from the design of the experiment that the data came from. In such cases, we should perform the `anova` test. It basically compares that model with the reduced model, without that factor out, and performs a likelihood ratio test. If the p-value is highly significant it means that the null hypothesis of having zero variance is rejected with assurance.

The choice of the link function is also important for goodness of fit. Here we have initially considered two different link functions `clog` and `cloglog` in model fitting and later stick with `cloglog` since this seemed more appropriate. It is noteworthy to mention that sometimes, depending on the data, applying a linear mixed model to a transformed data could make more sense and show superior results than that of a generalized linear mixed model on the original data.

Non-parametric random effect models could be a solution to the lack of fit of the model when the distribution of random effects is misspecified. Here we have used the `R` package `npmlreg` for non-parametric modeling of the random effects, which has shown adequate fitting and similar results to parametric version of the same model. In addition, having zero or very small random effect variance due to choosing inappropriate distribution is no longer a problem.

As further work, first of all, one could consider fitting generalized mixed effect models to a larger number of isolates, for a computationally better model fitting and to explore stability. Additionally, a Bayesian mixed effect model could be useful by setting prior distributions over the fixed effects. This could be implemented through `R` packages such as `blme`, `MCMCglmm` or `brms`.

4.7 Conclusions

The aim of this chapter was to establish a model that could capture the variability in the data in terms of modeling the mortality rate of the insects and capturing all important sources of variation. Different link functions giving different structures for the hazard model were considered with the newly defined `clog` link developed in `R`, since it is not included as an standard link function. Previous analyses of these data have not made use of such discrete survival models, instead focussing on the general ordinal multinomial structure together with random effects to account for the clustered structure and overdispersion. The advantage of the discrete survival model is that it focusses on the time to event nature of the data and the random effects models, in particular the non-parametric model, allow an (automatic) clustering of isolates with similar profiles.

Chapter 5

Survival Models Incorporating Genetic Profile

5.1 Introduction

Time to event, or survival, data is common in the biological and medical sciences with typical examples being time to death and time to recurrence of a tumour. In analyzing life-test data, the survival function, $S(t)$, is of paramount interest giving the probability of surviving at least to time t, $P(T \geq t)$, and providing crucial summary information from survival data for different values of t and the median time of survival, i.e. the survival time t that corresponds to probability of survival 0.5. Survival analysis techniques developed for dealing with censored survival data can be broadly stratified into three types: non-parametric (Kaplan-Meier/Product limit method), semi-parametric (Cox-proportional hazards method), and parametric methods, all of which are discussed briefly in this chapter. Survival analysis and the way of dealing with censoring has been discussed and explored by many authors, for e.g. Prinja et al. (2010), Clarke and Oxman (2000) and Efron and Hastie (2016). Note that we have already dealt with censored data in parametric ways in Chapter 3.

Regarding genetic information from a gene network, modules are highly connected subgraphs in a gene co-expression network that correspond to groups of genes having a similar function or involved in a common biological process which causes many interactions among themselves. In survival analysis, modular relations among gene expressions have been mostly ignored, although they carry great information that could improve statistical inference. It has been shown that using co-expressions and protein-protein interaction networks can remove the statistical randomness from high-dimensional data for several classification and regression models, see Chuang et al. (2007), Li and Li (2008), Hwang et al. (2008) and Zhang et al. (2013) for more information. These network based models have a better regularization across independent datasets as the network information is consistent with the conserved pattern of gene expressions. The most popular regularization (penalization) methods are Lasso, Ridge, LARs (least angle regression), elastic-net (a compromise between lasso and ridge) that do some form of variable selection and/or shrinkage. They penalize the model by maximizing the likelihood with respect to particular constraints that are required for estimation of parameters in modern large problems with many more predictor variables than observations, such as in high-throughput genomic data. We will briefly discuss these methods throughout the chapter. The main focus in this chapter is on the discussion of the gene network based models in survival analysis and comparing with competing models.

We start this chapter with some introductory concepts which are indeed important in interpretation of the data and prediction. We will then proceed with introducing methods for analyzing clinical and high dimensional genomic data. The methods are applied to a real data on ovarian carcinoma. Dynamic prediction of patients survival is obtained based on their clinical data and model based prediction is obtained based on their microarray gene expression data using the ridge regression variant of the so-called Cox regression.

5.2 Basic Survival Analysis Concepts

5.2.1 Survival function estimator

The Kaplan-Meier method, which is also called Product Limit Estimate, is a common non-parametric method to estimate the survival curve, an estimate of the probability of survival beyond time t, from the observed survival times (including the censored times). Now let X represent the typical lifetime. The estimate which is called the Empirical Survival Function, is given by

$$\tilde{S}(t) = P(X \geq t) = \frac{\text{number of individuals with lifetimes} \geq \text{t}}{\text{Total number of individuals}}. \tag{5.1}$$

Theoretically, the survival function can be graphed as a smooth curve; in practice, estimated versions based on finite samples give graphs that are step functions. The Kaplan-Meier method produces estimates of a table and a graph ($S(t)$ against t) that are referred to as the respective life table and survival curve. The shape of the survival curve depends upon the observed survival times and the proportion of censored observations.

Let $t_{(j)}, j = 1, \ldots, k$ be the j-th ordered observed time and n_j denote the number of individuals who are alive just before time $t_{(j)}$ and let d_j denote the number of deaths that occur at time $t_{(j)}$ (or, strictly within a small time interval from $t_{(j)} - \delta$ to $t_{(j)}$, $\delta \to 0$). Then the estimated probability of survival past $t_{(j)}$ is $\frac{n_j - d_j}{n_j}$. Assuming that the times $t_{(j)}$ are independent, the Kaplan-Meier estimate of the survivor function at time t is

$$\hat{S}(t) = \prod_{j=1}^{k} \left(1 - \hat{h}_j\right) = \prod_{j=1}^{k} \left(\frac{n_j - d_j}{n_j}\right) \tag{5.2}$$

for t between $t_{(j)}$ and $t_{(j+1)}$, where $\hat{h}_j = \frac{d_j}{n_j}$ is the estimator for *hazard rate* at time $t_{(j)}$ which by definition $h_j = P\{X = t_{(j)} | X \geq t_{(j)}\}$, i.e. the probability of dying at time $t_{(j)}$ given survival past $t_{(j-1)}$.

It is worth mentioning that similar to (5.2), we can define the probability S_{ik} of surviving past time $t_{(k)}$ given survival past time $t_{(i-1)}$ as the product of surviving each intermediate time,

$$\tilde{S}_{ik} = \prod_{j=i}^{k} \left(1 - \hat{h}_j\right) = P\left\{X \geq t_{(k)} | X \geq t_{(i)}\right\}. \tag{5.3}$$

This method is useful when there is no information about the parametric distribution of the hazard rate, however, we cannot generalize or predict the result beyond the study period, Swinscow et al. (2002).

To illustrate this, we use the ovarian dataset in the package `survival` in `R` to fit a Kaplan-Meier curve and plot it. This data shows the survival in a randomized trial comparing two treatments for ovarian cancer where `futime` is survival or censoring time, `fustat` is censoring status and `rx` is treatment group. We can find the Kaplan-Meier estimate as follows:

```
R> library(survival)
R> data(ovarian)
R> library(GGally)
# create a survival object
R> S <- Surv(ovarian$futime, ovarian$fustat)
R> S
[1]    59    115    156    421+   431    448+   464    475    477+   563
[11]   638    744+   769+   770+ 803+   855+ 1040+ 1106+ 1129+ 1206+
[21]    1227+   268    329    353    365    377+
```

```
R> fit1 <- with(ovarian, survfit(S ~ 1)) # fit a survival model
R> summary(fit1)
Call: survfit(formula = S ~ 1)

time n.risk n.event survival std.err lower 95% CI upper 95% CI
59       26        1     0.962  0.0377        0.890        1.000
115      25        1     0.923  0.0523        0.826        1.000
156      24        1     0.885  0.0627        0.770        1.000
268      23        1     0.846  0.0708        0.718        0.997
329      22        1     0.808  0.0773        0.670        0.974
353      21        1     0.769  0.0826        0.623        0.949
365      20        1     0.731  0.0870        0.579        0.923
431      17        1     0.688  0.0919        0.529        0.894
464      15        1     0.642  0.0965        0.478        0.862
475      14        1     0.596  0.0999        0.429        0.828
563      12        1     0.546  0.1032        0.377        0.791
638      11        1     0.497  0.1051        0.328        0.752
R> ggsurv(fit1)   # plot the survival curve

# change the breaks on the x- and y-axis
R> ggsurv(fit1)+scale_y_continuous(breaks = seq(0, 1, by = .1))
```

A plot of the survival curve is shown in Figure (5.1) where censored observations are denoted by a red cross and a confidence interval is generated. To create a confidence interval we need to find an estimator for the variance of the Kaplan-Meier estimator. This can be done by using the Greenwood's formula as follows:

According to (5.2),

$$\log(\hat{S}(t)) = \sum_{j=1}^{k} \log(1 - \hat{h}_j). \tag{5.4}$$

As $\hat{h}_j \sim \frac{1}{n_j} B(n_j, h_j)$, we have

$$\begin{aligned} \mathrm{var}\left\{\log(\hat{S}(t))\right\} &= \sum_{j=1}^{k} \mathrm{var}\left\{\log(1 - \hat{h}_j)\right\} \approxeq \sum_{j=1}^{k} \frac{\mathrm{var}(\hat{h}_j)}{(1 - h_j)^2} \\ &= \sum_{j=1}^{k} \frac{h_j}{1 - h_j} \frac{1}{n_j}, \qquad \text{since } \mathrm{var}(\hat{h_j}) = n_j\, h_j\, (1 - h_j) \end{aligned} \tag{5.5}$$

where we have used the delta method (2.13) for approximation of the variance with $\text{var}\left\{\log(X)\right\} \approxeq \text{var}(X)/E^2(X)$. Then if we substitute $\hat{h}_j = d_j/n_j$, we get

$$\widehat{\text{var}}\left\{\log(\hat{S}(t))\right\} \approxeq \sum_{j=1}^{k} \frac{d_j}{n_j(n_j - d_j)}. \tag{5.6}$$

Now that we have derived $\text{var}\left\{\log(\hat{S}(t))\right\}$, using the fact that $\text{var}(\hat{S}(t)) \approxeq E^2(\hat{S}(t)) \times \text{var}\left\{\log(\hat{S}(t))\right\}$ which gives the Greenwood's formula as follows:

$$\text{SD}(\hat{S}(t)) = \hat{S}(t)\left[\sum_{j=1}^{k} \frac{d_j}{n_j(n_j - d_j)}\right]^{1/2}, \tag{5.7}$$

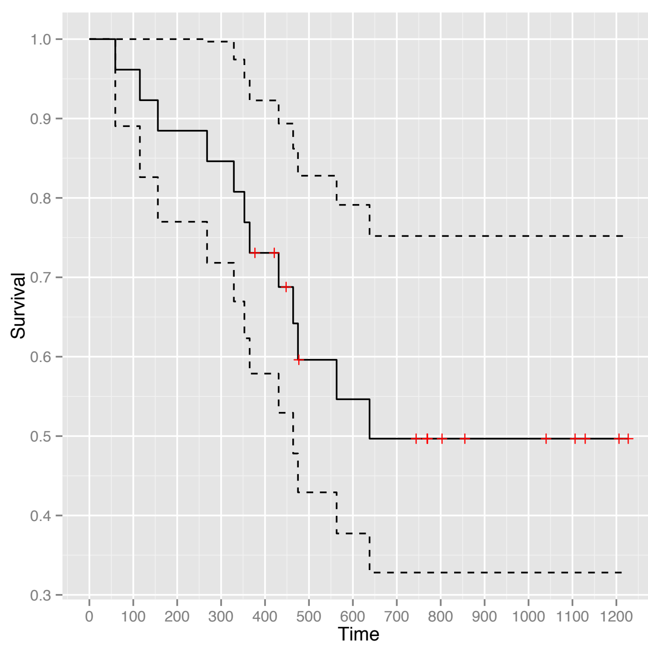


Figure 5.1: Kaplan-Meier estimate of survival function

```
R> fit2 <- with(ovarian, survfit(S ~ rx)) ## fit a survival model
R> summary(fit2)
Call: survfit(formula = S ~ rx)


rx=1
```

```
time n.risk n.event survival std.err lower 95% CI upper 95% CI
59     13       1    0.923  0.0739        0.789        1.000
115     12       1    0.846  0.1001        0.671        1.000
156     11       1    0.769  0.1169        0.571        1.000
268     10       1    0.692  0.1280        0.482        0.995
329      9       1    0.615  0.1349        0.400        0.946
431      8       1    0.538  0.1383        0.326        0.891
638      5       1    0.431  0.1467        0.221        0.840

rx=2
time n.risk n.event survival std.err lower 95% CI upper 95% CI
353     13       1    0.923  0.0739        0.789        1.000
365     12       1    0.846  0.1001        0.671        1.000
464      9       1    0.752  0.1256        0.542        1.000
475      8       1    0.658  0.1407        0.433        1.000
563      7       1    0.564  0.1488        0.336        0.946

R> ggsurv(fit2)  # plot the survival curve for both treatments

# change the breaks on the x- and y-axis
R> ggsurv(fit2) + scale_y_continuous(breaks = seq(0, 1, by = .1))
```

A plot of the survival curves is shown in Figure (5.2). It can be seen that patients on treatment 2 have a higher survival probability than those on treatment 1. Also we can estimate the median survival time (time with the probability of survival equal to 0.5) from this graph. The median survival time for patients on treatment 1 appears to be 640 whereas on treatment 2 the probability of survival is always higher than 0.5.

5.2.2 Dynamic Prediction Use of Survival Function

The Kaplan-Meier estimate of survival function discussed above, can be implied as a prognosis criterion for the patients at the starting time of the follow-up. In practice, the information given about the probability of patients survival is often based on evaluation of the estimated survival function at five years time, however, this might not be true as the clinical parameters of patients changes dynamically from time to time. Therefore, for a better answer to this question, we look at the conditional failure function mentioned in (2.14). This function can be rewritten by defining a fixed window of width w and considering the probability of failure within that window as a *fixed width failure function*, provided

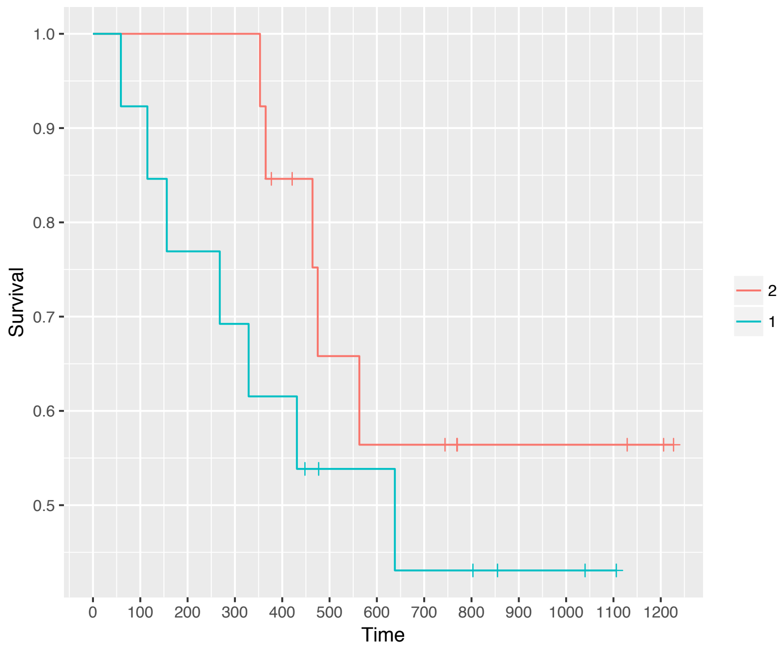


Figure 5.2: Ovarian cancer data: Kaplan-Meier estimate of survival for both treatments

that the patient is still alive at time t^{-}, as below.

$$F_w(t) = 1 - S(t + w|t)$$

where w is arbitrary and depends on the follow-up length and the overall prognosis. This function can be plotted using the `Fwindow` function in the `R` package `dynpred`, for the details on using this function, see van Houwelingen and Putter (2011). Figure 5.3 shows the probability of survival within the next five years from a dataset collected from a clinical trial in breast cancer patients, conducted by the European Organization for Research and Treatment of Cancer (EORTC trial 10854) with the aim of studying whether a short intensive course of perioperative chemotherapy gives better therapeutic results than surgery alone. For more detail about this data, see Putter et al. (2006) who has studied that.

Figure 5.3 illustrates that the good news for the patients is that the probability of death within the next 5 years drops dramatically from about 40% to about 10%. It shows that if is the patient is still alive and relapse-free after two years, their health has improved and there are good chances that the disease will not recur.

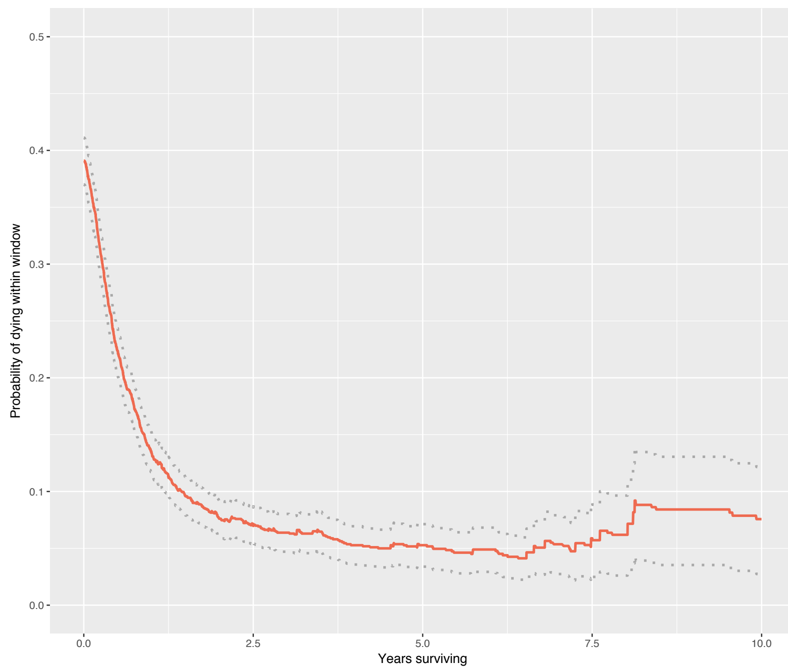


Figure 5.3: EORTC trial data: Probability of dying or relapse within the next 5 years with pointwise 95% confidence intervals

5.2.3 Log-rank Test

In clinical trials, interpretation of a two sample test is always important, e.g. for comparing the effect of a treatment on two groups of patients: treatment and placebo/control. This can be typically done by a simple two sample t-test or its nonparametric version Wilcoxon's test, however, these methods can not be employed when we have censored survival times. In this case, the *Log-Rank* test is a non-parametric method for comparing the two samples making use of the life tables that were mentioned in the previous section.

The idea behind this method is that, assuming there are N time periods it creates a two-by-two contingency table for time period i, $i = 1, \ldots, N$, of life tables so that the number of deaths can be considered conditionally on the treatment groups A and B as well as on the number of death and survivals. We form the log-rank test statistic

$$Z = \frac{\sum_{i=1}^{N} (d_i - E_i)}{\left(\sum_{i=1}^{N} V_i\right)^{1/2}}, \tag{5.8}$$

where d_i is the number of death in treatment A for period i, $E_i = E(d_i) = n_{Ai} n_{di}/n_i$ is the conditional expected number of deaths under treatment A for period i and $V_i = \text{var}(d_i) = \{n_{Ai}\; n_{Bi}\; n_{di}\; n_{si}/\left[n_i^2(n_i - 1)\right]\}$ is the conditional variance of number of deaths under treatment A for period i. This is illustrated in Table (5.1), where n_{Ai} is the num-

Table 5.1: Two-by-two display of data for time period i

	Died	Survived	
Treatment A	d_i	n_{12i}	$n_{Ai} = d_i + n_{12i}$
Treatment B	n_{21i}	n_{22i}	$n_B = n_{21i} + n_{22i}$
	$n_{di} = d_i + n_{21i}$	$n_{si} = n_{12i} + n_{22i}$	$n_i = n_{Ai} + n_{Bi}$

ber of people at risk in treatment A, n_{di} is the number of deaths, n_{si} is the number of survivors and n_i is the total number of people at risk in treatments A and B at time point i.

The null hypothesis is that the hazard rates are the same in both treatments, i.e,

$$H_0(i) : h_{Ai} = h_{Bi}, \quad \forall i \in \{1, \dots, N\}. \tag{5.9}$$

Therefore, overall we have the following hypotheses testing:

$$H_0 : h_A = h_B \qquad vs \qquad H_1 : h_A > h_B. \tag{5.10}$$

Under each individual hypothesis for each period, the numerator in (5.8) under $H_0(i)$ has the expectation zero and if h_{Ai} is greater than h_{Bi}, then it has a positive expectation showing the superiority of treatment B against the $H_0(i)$. Also, under H_0, $Z^2 \sim \chi^2_1$, where Z is as in (5.8).

Example 5.1. For the "`ovarian`" cancer study in Section (5.2.1), two treatments can be compared by using the log-rank test as follows.

```
R> survdiff(Surv(futime, fustat) ~ rx,data=ovarian)
# by default rho=0 doing log-rank test.

Call:
survdiff(formula = Surv(futime, fustat) ~ rx, data = ovarian)

     N   Observed  Expected (O-E)^2/E (O-E)^2/V
rx=1 13         7     5.23     0.596      1.06
rx=2 13         5     6.77     0.461      1.06

Chisq= 1.1  on 1 degrees of freedom, p= 0.303
```

Therefore, the null hypothesis of equal hazard rates in both treatments is not rejected and there is evidence to show that both treatments are acting quite similarly.

5.3 Cox proportional hazards model

In cancer genomics and analyzing microarray gene expression data, a good survival model is expected to identify genes that are predictive of the events such as recurrence of a tumor. In practice, survival data is typically subject to censoring with incomplete observation of some failure times due to drop-out, intermittent follow-up and finite study duration.

Among the widely-used functions in survival analysis: survival function, distribution function, hazard function and the density function describing the survival times, the hazard function is the one that captures the aging process the most. A fully parametric hazard function can describe two things: (1) the distribution of the survival times (error component), (2) how that distribution changes as a function of possible covariates (systematic component). In some practical settings, it is important to use a model that takes both these two into account, however, in other settings, when the goal of the statistical analysis is estimating parameters to describe the survival times in a comparative sense (e.g. between two groups of patients), such as how survival time is affected by the treatments under study, a model that only describes the latter is all that is necessary, that is, the assumptions for the error component may be unnecessary when the inference is just on the parameters in the systematic part of the model. These models are called semi-parametric models as the underlying survival time distribution is not known or specified.

The Cox proportional hazards model, proposed by Cox (1972), is commonly used to predict the outcome of interest using the time to an event of interest. It is a semi-parametric multiple regression model that allows inclusion of explanatory variables while testing the difference between survival times of different groups of patients, which may help improve the precision of estimation of the treatment effect. It can also be used to analyze the survival of the patients in a clinical trial by isolating the effect of treatment from the effect of other explanatory variables. The response variable is the hazard function which is the instantaneous rate of patient's death. This model which is also known as the Cox regression is defined as follows:

$$\ln \frac{h(t)}{h_0(t)} = \beta_1 x_1 + \cdots + \beta_p x_p, \tag{5.11}$$

where $h(t)$ is the hazard function at time t, $x_1, x_2, \ldots, x_p$ are the explanatory variables for an individual observation, $h_0(\cdot)$ is the baseline hazard when all the explanatory variables are zero and $\beta_1, \beta_2, \ldots, \beta_p$ are the regression coefficients. There is no assumption about the distribution of the hazard function, however, the hazard ratio $\frac{h(t)}{h_0(t)}$ is independent of time, i.e. the ratio of hazards from one group to that of the other group at a particular time remains the same over time.

In any regression model, the estimated coefficient shows the rate of change in the (function of) dependent variable given a unit change in a covariate. In proportional hazards model, this shows the effect of one or more covariates on survival times. Therefore, a correct interpretation of coefficients is important in the study of the effect of covariates.

Assume $p = 1$ with the above parametrization and $\beta_1 = \beta$ in (5.11). The hazard ratio for two individuals with covariate values a and b is $\mathrm{e}^{\beta(b-a)}$. For instance, if the value of the coefficient β is equal to $\ln 2$ and b is one unit larger than a, it is interpreted as the patient with covariate value b is dying at the twice rate of the other patient. Note that, as the baseline hazard function, $h_0(t)$, is a component in the hazard function for both covariates, they cancel out in calculations. Also the hazard ratio does not depend on time as mentioned before.

More specifically talking about the model (5.11), given $\boldsymbol{X}$ is the vector of p covariates/risk factor, for e.g. the gene expression profile of n patients over p genes, the instantaneous risk of an event occurring at time t for the i-th patient with gene expressions $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip})'$, $i = 1, \ldots, n$, can be written as:

$$h(t|\boldsymbol{X}_i) = h_0(t) \exp(\boldsymbol{X}_i' \, \boldsymbol{\beta}), \tag{5.12}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)'$ is the vector of regression coefficients that we want to estimate.

The Survival Function derived from this model is given by:

$$S(t|\boldsymbol{X}) = \exp\big(-\exp(\boldsymbol{X}'\boldsymbol{\beta})H_0(t)\big) = S_0(t)^{\exp(\boldsymbol{X}'\boldsymbol{\beta})},$$

where $H_0(t) = \int_0^t h_0(s)\,ds$ is the *cumulative baseline hazard* and $S_0(t) = \exp\big(-H_0(t)\big)$ is the baseline survival function.

The linear predictor $\boldsymbol{X}'\boldsymbol{\beta}$ from the Cox model (5.12) is called the ***Prognostic Index***. The prognostic index is an important factor which can be used to describe low-risk and high-risk groups of patients and also for dynamic prediction of patients survival times. The use of the prognostic index obtained from both clinical and genomic data is discussed in more detail later. We denote this as PI in the rest of this chapter.

In semi-parametric models, to avoid specifying the distribution of the survival time, Cox's partial likelihood, which is only dependent on the parameters of interest, was proposed by Cox (1972) since the log-likelihood function is not available to use for statistical inference. It is shown that the parameter estimates and statistical tests results obtained based on the partial likelihood function have the same distributional properties as the maximum likelihood estimators from the fully parametric likelihood. For the detailed discussion of this, see Hosmer Jr and Lemeshow (1999) and Kalbfleisch and Prentice (2011) and Collett (2015).

Let t_i be the observed/censored survival time for the i-th patient and $R(t_i)$ be the risk set at time t_i, i.e. the set of patients with survival time (observed or censored) greater than or equal to the specified time t_i, that is,

$$R(t_i) = \{j : t_j \geq t_i\} \tag{5.13}$$

Lemma 5.1. *Under the proportional hazards model (5.12), the probability that individual j in the risk set $R(t_i)$ is the one that dies at time t_i, conditional on the risk set $R(t_i)$, is given by*

$$\mathrm{P}\{j \textit{ dies at } t_i | R(t_i)\} = \frac{\mathrm{e}^{\boldsymbol{X'}_j \beta}}{\sum_{k \in R(t_i)} \mathrm{e}^{\boldsymbol{X'}_k \beta}}. \tag{5.14}$$

With the current settings and assuming there are no ties among the t_i's, the partial likelihood function is then defined as the product of factors (5.14) and is given by:

$$L_p(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left[\frac{\exp\left(\boldsymbol{X}_i' \ \boldsymbol{\beta}\right)}{\sum_{j \in R(t_i)} \exp(\boldsymbol{X}_j' \ \boldsymbol{\beta})} \right]^{\delta_i} \tag{5.15}$$

where d_i is the censoring indicator i.e. $\delta_i = 0$ if t_i is censored and 1 if it is observed. The partial log-likelihood, pl, is given by

$$pl(\boldsymbol{\beta}) = \sum_{i=1}^{n} \delta_i \left\{ \boldsymbol{X}_i' \ \boldsymbol{\beta} - \ln \left[\sum_{j \in R(t_i)} \exp\left(\boldsymbol{X}_j' \ \boldsymbol{\beta} \right) \right] \right\} \tag{5.16}$$

which is maximized for the estimation of the unknown parameters $\boldsymbol{\beta}$.

Remark. *Notice that equation (5.15) can be modified by excluding the terms corresponding to δ_i=0 which gives the following formula:*

$$L_p(\boldsymbol{\beta}) = \prod_{i=1}^{m} \frac{\exp\left(\boldsymbol{X}_{(i)}' \ \boldsymbol{\beta}\right)}{\sum_{j \in R(t_{(i)})} \exp(\boldsymbol{X}_j' \ \boldsymbol{\beta})}$$

where $t_{(i)}, i = 1, 2, \ldots, m$ are the m ordered survival times, i.e. there are m observed deaths.

Given the estimated regression coefficients $\boldsymbol{\beta}$, we can estimate the baseline hazard function by the Breslow estimator, Breslow (1972), given by

$$\hat{h}_0(t_i) = \frac{1}{\sum_{t_j \geq t_i} \exp\left(\boldsymbol{X}'_j\, \hat{\boldsymbol{\beta}}\right)} = \frac{1}{\sum_{j \in R(t_{(i)})} \exp\left(\boldsymbol{X}'_j\, \hat{\boldsymbol{\beta}}\right)} \tag{5.17}$$

The partial likelihood and the Breslow estimator can both be derived from the total log-likelihood

$$l(\boldsymbol{\beta}, h_0) = \sum_{i=1}^{n} \left[- \exp\left(\boldsymbol{X}'_i\right)\; H_0(t_i) + d_i(\ln(h_0(t_i)) + \boldsymbol{X}'_i \boldsymbol{\beta}) \right] \tag{5.18}$$

with $H_0(t) = \sum_{s \leq t} h_0(s)$.

That is to say that (5.16) is the profile log-likelihood for the parameter vector $\boldsymbol{\beta}$ corresponding to (5.18) and maximizing (5.18) with respect to $h_0(t)$ gives (5.17). The estimation of $\boldsymbol{\beta}$ and $h_0(s)$ is usually done through an alternating optimization algorithm for $\boldsymbol{\beta}$ and $h_0(s)$, van Houwelingen et al. (2006). The maximization with respect to $\boldsymbol{\beta}$ is done by Newton-Raphson and the optimized value of $h_0(s)$ can be found from (5.17). Zhang et al. (2013) and van Houwelingen et al. (2006) used (5.16) for this purpose and van Houwelingen et al. (2006) found it easier than doing that using the partial likelihood pl in terms of being computationally faster and with no need for handling ties, (van Houwelingen et al., 2006, page 3205).

In classical settings, when $n > p$, in order to obtain the parameter estimates using a maximum likelihood approach, we have p equations, corresponding to derivatives of the log-partial likelihood with respect to the parameters for each covariate, and we set them equal to zero and solve, which results in the maximum partial likelihood estimators. The estimator for the standard error of the parameter estimates can also be easily found and is equal to the square root of the inverse of the negative second derivative of the partial log-likelihood pl, the observed information matrix. To this end, the Cox model is relatively easy to implement because of its intuitive likelihood modeling for both censored and observed patient samples who are event free by the last follow-up.

It has been shown that existing Cox models can be trained just for a particular dataset and cannot be generalized properly to other independent datasets. For details see Zhang et al. (2013) and Walters (2012).

5.3.1 Regularization methods

Nowadays with the growing appreciation of genomic data and the variety of biomedical applications of gene expression microarrays, classical survival analysis has been integrated with the analysis of high dimensional gene expression profiles of patients. However, in analysis of high-throughput data, such as the microarray or RNA-seq gene expression data, the number of gene features p is much larger than the number of individuals ($p >> n$), hence, due to high dimensionality of genomic data the use of the Cox regression model for survival analysis is subject to some regularization with applying penalties over the partial log-likelihood, or dimension reduction principal components. The penalization methods include the L2 penalty as in ridge regression, L1 Lasso regularization, Elastic-net, etc., which are designed to improve the high dimensionality and overfitting problem, although Zhang et al. (2013) showed that the Cox model with a small sample size is still likely to overfit. We briefly explain some popular regularization methods. For more detailed explanation one could refer to Hastie and Friedman (2009).

Ridge Regression

Ridge regression is a shrinkage method that was first introduced by Hoerl and Kennard (1970) as a method for modifying the ordinary least squares regression for continuous outcomes in the presence of multicollinearity. This method basically uses the idea of penalizing the residual sum of squares by the sum-of-squares of the parameters (L2- penalty). In other words, it shrinks the regression coefficients by applying a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares,

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{5.19}$$

where $\lambda \geq 0$, $(y_1, \ldots, y_n)$ are observations and x_{ij} is the j-th covariate associated with the i-th observation, (Hastie and Friedman, 2009).

The parameter λ in (5.19) is a penalty parameter that controls the amount of shrinkage - the higher the value of λ, the higher the amount of shrinkage. Note that the shrinkage in parameters is made towards zero and towards each other.

The log-likelihood for ridge version of the Cox model is defined as follows by applying the L2 penalty on the total likelihood,

$$l_{\text{pen}}(\boldsymbol{\beta}, h_0) = l(\boldsymbol{\beta}, h_0) - \frac{1}{2}\lambda\boldsymbol{\beta}'\boldsymbol{\beta}, \quad \lambda \geq 0 \tag{5.20}$$

or equivalently applying on the partial log-likelihood,

$$pl_{\text{pen}}(\boldsymbol{\beta}) = pl(\boldsymbol{\beta}) - \frac{1}{2}\lambda\boldsymbol{\beta}'\boldsymbol{\beta} \tag{5.21}$$

where pl_{pen} denotes the penalized partial log-likelihood and $l(\boldsymbol{\beta}, h_0)$ and $pl(\boldsymbol{\beta})$ are defined respectively in (5.18) and (5.16).
Notice that although both (5.20) and (5.21) lead to the same results for $\boldsymbol{\beta}$, the penalized total likelihood depends upon the baseline hazard while the penalized partial likelihood does not.

Lasso Regression

Lasso is an innovative variable selection method for regression. Variable selection in regression is extremely important when we have available a large collection of possible covariates from which we hope to select a parsimonious set for the efficient prediction of a response variable. Lasso is a shrinkage method like ridge which minimizes the residual sum of squares but subject to the sum of the absolute value of the coefficients being less than a constant (L1-penalty), rather than the sum-of-squares of the parameters (L2-penalty). It not only helps to improve the prediction accuracy when dealing with multicolinearity data, but also carries several nice properties such as interpretability and numerical stability. The lasso estimate is defined by

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \right\}, \quad \text{subject to} \sum_{j=1}^{p} |\beta_j| \leq s \tag{5.22}$$

where s is a tuning parameter. Choosing s is like choosing the number of predictors to use in a regression model, and cross-validation is a good tool for estimating the best value for s. Notice that when s is large enough, the constraint has no effect and the solution can be found in the same way as in a usual multiple linear regression. (5.22) is equivalent to

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{5.23}$$

LARS (Least Angle Regression and Shrinkage) is an efficient stepwise variable selection algorithm and can be modified to compute the regularization path of the lasso. With the `lasso` option in the `lars` function in `R`, it computes the complete lasso solution simultaneously for ALL values of the shrinkage parameter at the same computational cost as a least squares fit. LARS is described in detail in Efron et al. (2004).

A Simple Explanation of Lasso and LARS

Given a set of input measurements $x_1, x_2 ... x_p$ and an outcome measurement y, the lasso fits a linear model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p,$$

and the Least Angle Regression algorithm is as follows:

- Start with all coefficients β_j equal to zero.
- Find the predictor x_j most correlated with y
- Increase the coefficient β_j in the direction of the sign of its correlation with y. Take residuals $r = y - \hat{y}$ along the way. Stop when some other predictor x_k has as much correlation with r as x_j has.
- Increase (β_j, β_k) in their joint least squares direction, until some other predictor x_m has as much correlation with the residual r.
- Continue until: all predictors are in the model

It can be shown that, with one modification, this procedure gives the entire path of lasso solutions, as s is varied from 0 to infinity. The required modification is: if a non-zero coefficient hits zero, remove it from the active set of predictors and recompute the joint direction.

(Note that the above is from Rob Tibshirani's web page on Stanford University website. For more details, see `http://statweb.stanford.edu/~tibs/research\_page.html`.)

Elastic-net

In elastic-net regularization in the framework of survival analysis, the goal is to maximize the partial likelihood with respect to the constraint,

$$\sum_{j=1}^{p} \Big(\alpha\beta_j^2 + (1-\alpha)|\beta_j| \Big) \leq c.$$

The elastic-net is a compromise between the lasso and ridge combining the strong points of each. The lasso penalty tends to pick a few non-zero predictors (which is also known as a variable selection method) and if the two predictors are very correlated, it throws away one and only keeps one of them, while the ridge penalty method shrinks all predictor coefficients towards zero and not exactly zero which helps for the high dimensional data when $p >> n$, in the sense of assigning a coefficient to each covariate (gene). Moreover, the ridge penalty has a better capability in handling correlated data as if two predictors are highly correlated it gives them equal weights. For more theoretical details on elastic-net, see Simon et al. (2011) and Hastie and Friedman (2009).

As an overall comparison between the three penalization methods mentioned above, Ridge regression shrinks the coefficients proportionally while Lasso shrinks most of them towards exactly zero which is known as "soft thresholding" in machine learning. Notice that for a fixed value of λ, elastic-net with $\alpha = 1$ give similar results as lasso and as α tends to 0, the results become similar to ridge type, which makes sense based on their associated defined constraints. In this chapter we use the ridge regression as a penalized method for high dimensional genomic data.

5.4 Assessing the prediction models

A generic way of measuring the quality of a parametric model is by looking at the likelihood of the collection of fitted models, for example from a very simple model, as a null model, to a very complex model, as a saturated model. The difference between the likelihood of the simpler model and the more complex model can be used for the purpose of assessing the model performance, which is basically what the likelihood ratio test with the chi-squared distributed test statistic (degrees of freedom equal to the increase in the number of parameters), defined as twice the difference in log-likelihoods, does. The performance of a predictive model can also be assessed in a similar way. Akaike (1974) has shown that the optimism, that is the expected difference between the fitted log-likelihood and the actual log-likelihood in new observations from the same individuals, is equal to the dimension of the model, that is the numbers of free parameters in the model. This error measure is extensively used in parametric models, however, is not feasible for a semi-parametric model such as Cox model, but it could still be used on the Cox partial log-likelihood which is free of the baseline hazard. Notice that using a new observation with its own new event time is not possible in the estimated Cox model, however, the partial log-likelihood is a measure of how well a model can distinguish the patients according to their survival time and is defined for the whole observations and not a single one. Therefore, for this purpose, the cross validation method is introduced, as explained in the next section, which interestingly does not need new data for this assessment.

5.4.1 Cross-validation

Cross-validation is a practical way of measuring the goodness of predictive models, inspired from the so-called *split sample* approach, where the data are split into training and test sets and the performance of the model is checked by predicting the test set using the training set(s). When the data are split into k datasets of roughly the same size, this is called k-fold cross-validation. However, the result might depend on the cut-off splitting points. Therefore, later in this chapter we use the leave-one-out cross-validation which refers to the case that k is equal to the number of observations and stays closest to the actual partial log-likelihood. Cross validation is also useful to deal with overfitting when the same data is used for both modeling and assessing the predictive performance.

The cross-validated partial log-likelihood is defined as

$$\text{CVPL} = \sum_{i=1}^{n} \left(pl(\hat{\boldsymbol{\beta}}_{\lambda}^{(-i)}) - pl^{(-i)}(\hat{\boldsymbol{\beta}}_{\lambda}^{(-i)}) \right) \tag{5.24}$$

where $\hat{\boldsymbol{\beta}}_{\lambda}^{(-i)}$ is the leave-one-out regression coefficients estimate which maximizes the leave-one-out partial log-likelihood $pl^{(-i)}\,(\boldsymbol{\beta})$ for a given value of λ; pl is defined in (5.16). The optimal value of the penalty parameter λ is then found by maximizing the cross-validated partial log-likelihood (5.24) with respect to λ. Notice that $pl(\boldsymbol{\beta}) - pl^{(-i)}(\boldsymbol{\beta})$ in (5.24) represents the "independent" contribution of individual i to the partial log-likelihood.

5.5 Application to Ovarian Carcinoma Data

Ovarian cancer is one of the leading causes of death in women in recent years with most deaths for patients with advanced-stage, high-grade serous ovarian cancer (HGS-OvCa) as reported in Koonings et al. (1989) and Seidman et al. (2004). The standard treatment is aggressive surgery followed by platinumtaxane chemotherapy. After therapy, platinumresistant cancer recurs in approximately 25% of patients within six months, and the overall five-year survival probability is 31%. The Cancer Genome Atlas (TCGA) provided researchers a possibility to study comprehensively genomic and epigenomic abnormalities on clinically annotated HGS-OvCa samples. In this study the messenger RNA (mRNA) expression, microRNA expression, promoter methylation and DNA copy number in 489 high-grade serous ovarian adenocarcinomas and the DNA sequences of exons from coding genes in 316 of these tumours are measured.

In Cancer Genome Atlas projects gene expressions of the samples are measured multiple times on different microarray platforms. However, the unified expression is defined as a weighted average of the platforms, and performs well in terms of accuracy and precision. Also, the advantages of using the unified gene expression is firstly working with only one set of measurements rather than multiple ones, and more importantly the estimated expression levels are more accurate and precise, see Wang et al. (2011) for more detail. We have used the complete-data unified gene expression profile of patients and their associated clinical data in this section, which consists of 269 patients gene profile with 11864 employed genes, to build a predictive model for patients' survival and interpret how effective the treatment is for the patients in high-risk and low-risk prognosis groups.

The data are summarized as a rich data in the class of `ExpressionSet` using the BioConductor's `BioBase` package - an `ExpressionSet` is a standard format in Bioconductor for storing gene expression data and is the input, or output, from many Bioconductor functions. This data consists of three components: (i) assayData, which explain different methods used for accessing different components of the expression data. (ii) phenoData, consists of meta-data which describes the samples in the experiment. (iii) Annotation, which is a character string identifying the type of chip used in the microarray experiment. Each of these components are created and then assembled in a rich *ExpressionSet* data. A summary of the data is shown below.

```
R> rich_data
ExpressionSet (storageMode: lockedEnvironment)
assayData: 11864 features, 269 samples
element names: exprs
phenoData
sampleNames: TCGA-04-1331 TCGA-04-1347 ... TCGA-61-2101 (269 total)
varLabels: AgeAtDiagnosis..yrs. VITALSTATUS ... risk4_fact (16 total)
varMetadata: labelDescription
Annotation: HGS-OvCa
```

First off, we start with analyzing the clinical data followed by analyzing the genomic data.

5.5.1 Clinical Data Analysis

The clinical data includes the measurements of 12 clinical variable/risk factors. As some insight to the clinical data, the data structure is given below. The variable age at which a patient is diagnosed with cancer is categorized into groups of below and over age 50, for the ease of interpretation and better model fit, and is named as "`age_c`".

```
R> str(ov_cl3)
'data.frame': 269 obs. of  13 variables:
$ AgeAtDiagnosis..yrs.        : num  79 81.4 44.5 61.1 50.9 ...
$ VITALSTATUS                 : int  1 2 1 1 2 1 1 1 1 1 ...
$ TUMORSTAGE                  : Factor w/ 7 levels "IIA","IIB","IIC",..: 6 7 5 6 6 4 6 6 6 5 ...
```

```
$ TUMORGRADE                 : Factor w/ 2 levels "G2","G3": 2 2 2 2 2 1 2 2 2 1 ...
$ TUMORRESIDUALDISEASE       : Factor w/ 4 levels ">20mm","1-10 mm",..: 2 4 2 2 4 4 1 2 2 2 ...
$ PRIMARYTHERAPYOUTCOMESUCCESS: Factor w/ 4 levels "COMPLETE RESPONSE",..: 1 1 1 3 1 1 1 1 1 1 ...
$ PERSONNEOPLASMCANCERSTATUS  : Factor w/ 2 levels "TUMOR FREE","WITH TUMOR": 2 1 2 2 1 1 2 2 2 2 ...
$ ProgressionFreeStatus      : Factor w/ 2 levels "DiseaseFree",..: 2 1 2 2 1 2 2 2 2 2 ...
$ PlatinumStatus             : Factor w/ 4 levels "Missing","Resistant",..: 3 3 3 2 3 3 3 2 3 3 ...
$ OverallSurvival_mo         : num  43.8 62.9 48.6 33.6 71.3 ...
$ PlatinumFreeInterval_mo    : num  9.6 57.4 12.8 3.3 64.9 34 6.7 0.9 6.4 27.6 ...
$ ProgressionFreeSurvival_mo : int  67 269 102 323 292 217 27 266 28 194 ...
$ age_c                      : Factor w/ 2 levels "<50",">50": 2 2 1 2 2 1 2 1 2 2 ...
```

As the main aim is to predict the patients survival, we first fit a multivariate Cox proportional hazards model. This is summarized in Table 5.2 below.

Table 5.2: Ovarian Carcinoma Data: The fitted Cox model

Covariate	Category	B	SE
Age	<50		
	> 50	-0.53045	0.20693
PRIMARYTHERAPYOUTCOMESUCCESS	COMPLETE RESPONSE		
	PARTIAL RESPONSE	-0.03624	0.37768
	PROGRESSIVE DISEASE	-0.36784	1.02336
	STABLE DISEASE	1.60702	0.42963
PERSONNEOPLASMCANCERSTATUS	WITH TUMOR	-0.02635	0.43671
	TUMOR FREE		
TUMOR RESIDUAL DISEASE	1-10 mm	0.09371	0.27738
	11-20 mm	-0.16016	0.52664
	No Macroscopic disease	0.23343	0.29773
	>20mm		
TUMORSTAGE	IIA		
	IIB	2.68150	1.47343
	IIC	1.16072	1.10828
	IIIA	0.95547	1.45071
	IIIB	0.51499	1.18636
	IIIC	0.73912	1.06495
	IV	0.93390	1.08804
TUMORGRADE	G2		
	G3	0.48498	0.32463
Progression Free Status	Disease Free		
	Recurred/Progressed	-2.08086	0.44874
Progression Free Survival	Continuous	-0.00256	0.00121

The relevant information is summarized in the PI $= (\boldsymbol{X} - \overline{\boldsymbol{X}})'\hat{\boldsymbol{\beta}}$ which is scale free version of the linear predictor. It is also independent of centering and categorizing the variables and if the value of covariates change the parameter estimates $\hat{\boldsymbol{\beta}}$ will change too. The overall mean of PI is zero and its standard deviation is 1.052. PI can be considered as a risk measure, i.e. the higher the value of PI, the higher risk of death. Therefore, the distribution of PI shifts to left over time which can be seen from Figure 5.4. To be more specific, one can look at the mean and standard deviation of PI in the group of patients at risk after four years and those who are no longer at risk within four years of the follow-up start date. The mean and standard deviation of the first group are -0.27 and 0.77 and those of the second group are 0.11 and 1.13.

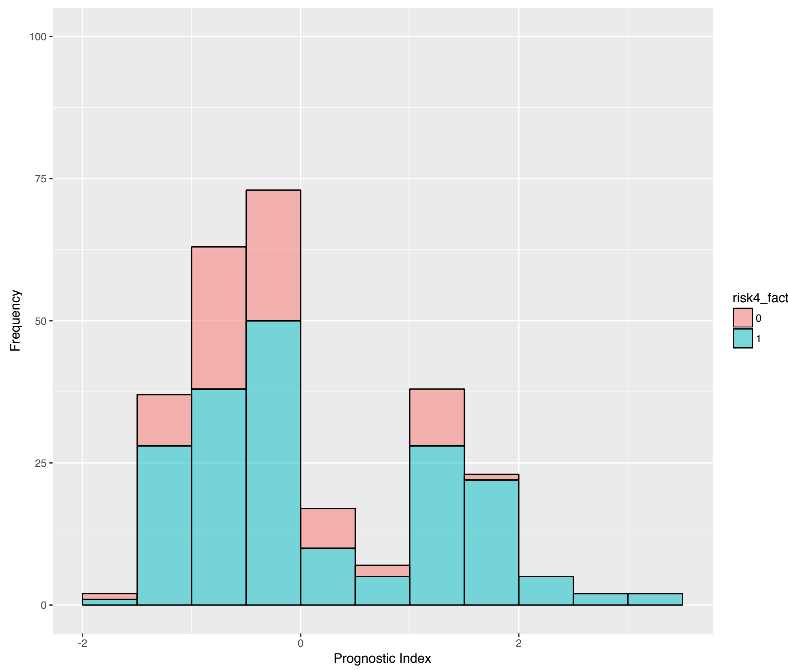


Figure 5.4: Histogram of the prognostic index PI

As mentioned before, the survival curve is a useful tool for obtaining the probability of survival for patients with poor and good prognosis. Now, we plot the survival curve for different values of the PI (within the range of $\overline{\text{PI}}$ − 2 SD (PI) to $\overline{\text{PI}}$ + 2 SD (PI)). The estimated survival probability for each value of PI can be found from

$$S(t|PI) = S_0(t)^{\exp(PI)}, \tag{5.25}$$

and is shown in Figure 5.5. This can be done in `R` by rerunning a Cox model with the obtained PI as a single predictor – its estimated coefficient is exactly one, indicating a good calibration, as the same data has been used for both model fitting and assessing the prediction performance in the second Cox model.

Survival probability estimates for over 10 years (> 125 months) in this figure for different values of the PI (from Mean − 2 SD to Mean + 2 SD), vary from zero to 62%. For the group of patients with the worst prognosis the probability of survival drops rapidly whereas for the best prognosis group (with lowest PI values) the probability of survival stays above 60% and within more than 10 years it reduces by the amount of about 0.375. The overall survival (by the Kaplan-Meier estimate) after ten years drops from one to about 8%.

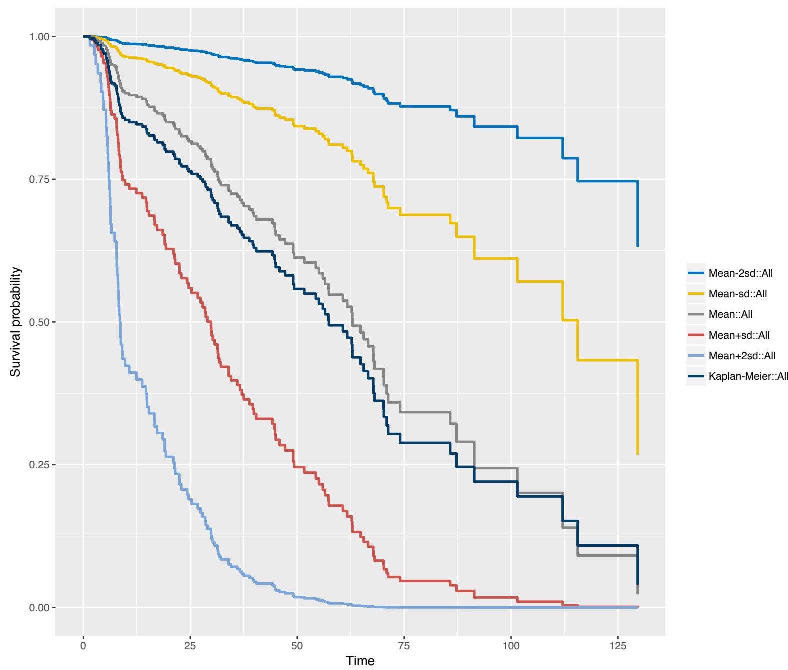


Figure 5.5: Predicted survival curves for different values of the PI

Dynamic prediction

As described before in Section 5.2.2 of this chapter, for dynamic prediction of survival within a fixed window, we look at the dynamic effect of the PI, which can be obtained from the `Fwindow` function in `R`. Figure 5.6 pictures the predicted probability of dying within a window of five years plotted for the first eight years, for the same values of PI as in Figure 5.5.

As we can see from the plot, the probability of dying within the next five years for the low-risk patients stays near zero at any point in the first eight years. For the high-risk group, at the start of the treatment the probability of dying increases by about 30% until four years or so, and afterwards the risk of death reduces to a probability of about 0.3 eight years after start of the treatment. For the other groups we see that after four years, there is a slight increase of probability of death, however, it gradually converges to a nearly constant probability. This means that the good news is that the treatment

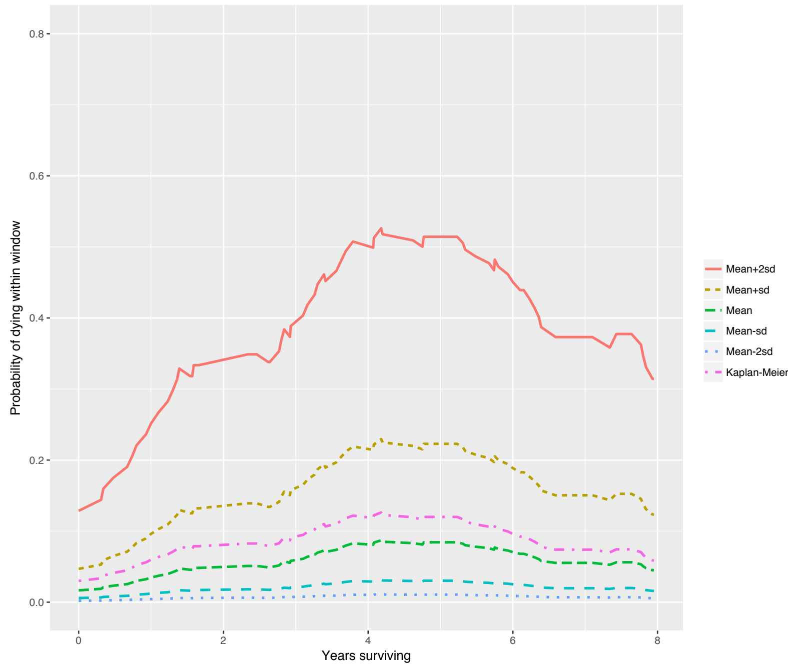


Figure 5.6: Dynamic prediction curves for different values of the PI

is advantageous for at least the good prognosis group of patients, although for high-risk patients the outcome is not good.

5.5.2 Genomic Data Analysis

The first screening step in building a predictive model based on the gene expression data is conducting a *globat test*. The global test checks the overall association of survival and gene expression data, very important in integrated analysis of clinical and microarray gene expression data set, and the predictive significance, i.e. it tests the null hypothesis of whether none of the genes are effective. If the test result is not significant, no predictive model can be fit and an alternative method should be considered, see Goeman et al. (2005) and van Houwelingen and Putter (2011) for more theoretical details. The summary of the global test for the HGS-OvCa data is given in Table 5.3.

Table 5.3: Global test

p-value	Statistic	Expected	Std.dev	#Cov
0.00275	0.805	0.372	0.156	11864

As the p-value (0.00275) is highly significant, it assures that prediction based on a penalized likelihood can be useful. Furthermore, functions `covariates` or `features` from the `globaltest` package give the diagnosis plot in Figure 5.7 which plots the p-values of testing genes. This plot can be used as a visual exploration of which genes or groups of genes are similar in terms of their expression signature.

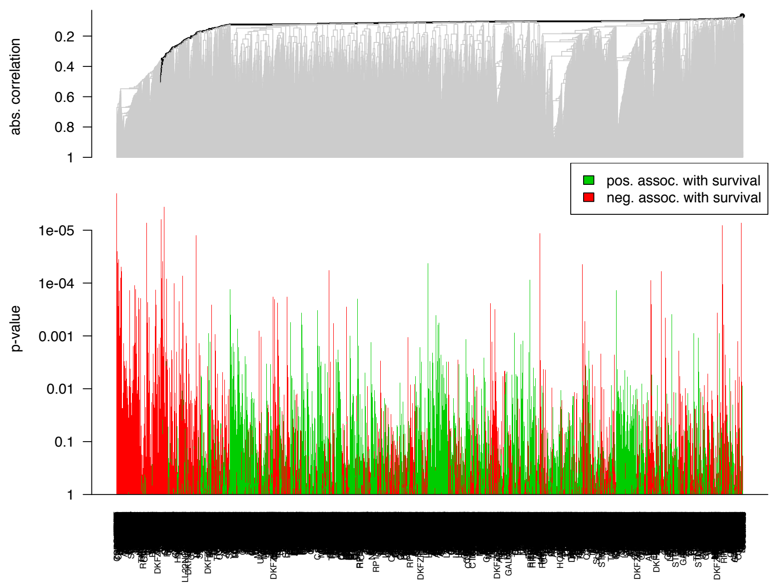


Figure 5.7: Features plot

Univariate Cox regression (one single gene at a time) shows that 19% of the fitted coefficients are significant at the 5% level. The Funnel plot in Figure 5.8 also shows that the great majority of coefficients lie within the bounds of -2 SE and 2 SE as a function of the information on them ($1/\mathrm{SE}^2$) and so there are not so many genes with outstanding effects.

Gene Expression Differentiation

Identification of differential gene expression is the first task of an in depth microarray analysis which detects genes whose patterns of expression differ according to phenotype or experimental conditions. Identifying differentially expressed genes using either microarrays or RNA-seq, which is not our focus here, is one of the most common objectives among investigators. For example, it might be of interest to discover transcripts which show different average expression levels across two groups of experiments: normal and diseased.

The approach for finding differential expressions in microarrays is simple: The measurements obtained from the same probe are compared across samples, however, different

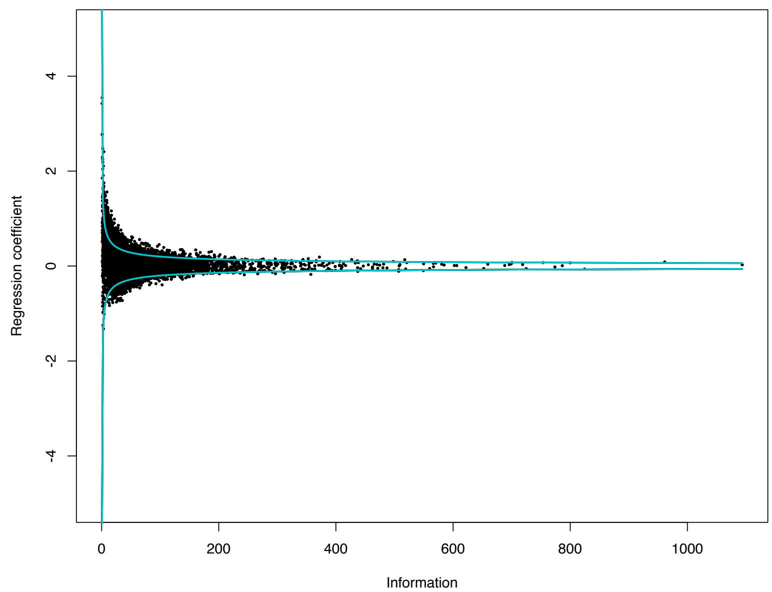


Figure 5.8: Funnel plot

summarization approaches can lead to very different statistical inference.

The `limma` Bioconductor package provides functions `topTable` and `decideTests`, which summarize the results of the linear model, conduct hypothesis tests and performs the multiple testing adjustment for the p-values. Results include (log) fold changes, standard errors, t-statistic and p-values.

The top ten differentially expressed genes between patients with progression status "Disease Free" and those with "Recurred/Progressed" status, are summarized in Table 5.4 using the `topTable` function, and the heatmap of those is shown in Figure 5.9. The summary statistics are presented in the table. The value of contrast usually indicates a $\log_2$-fold change between experimental conditions although sometimes it shows a $\log_2$-expression level. Note that formerly, simple "fold change approach was used to find differences under the assumption that changes beyond some threshold value were significant biologically, for example, two-fold.

The basic statistic used for significance analysis is the *moderated t-statistic*, which is computed for each contrast. The `AveExpr` column is the average $\log_2$-expression level for that gene across all the arrays in the experiment. The Next column (column t), is the value of the moderated t-statistic. Moreover, column adj.P.Val is the associated adjusted p-value after multiple testing adjustment. The most common form of adjustment and that used here is "`fdr`", which is also known as Benjamini and Hochberg's method to control the false discovery rate, see Benjamini and Hochberg (1995). The final column is B-statistic being the log-odds that the gene is differentially expressed.

Table 5.4: Top ten differentially expressed genes

	logFC	AveExpr	t	P.Value	adj.P.Val	B
SIX5	0.246	-0.010	4.267	2.731e-05	0.324	1.118
MYO16	-0.401	-0.005	-4.042	6.893e-05	0.408	0.488
MRPS11	-0.275	0.042	-3.876	1.328e-04	0.525	0.0438
TUBA1B	-0.235	0.066	-3.622	3.479e-04	0.578	-0.606
C8ORF70	0.391	-0.016	3.505	5.335e-04	0.578	-0.893
MRPL16	-0.228	0.025	-3.448	6.537e-04	0.578	-1.030
CCR7	-0.408	-0.081	-3.443	6.646e-04	0.578	-1.041
FEN1	-0.290	0.065	-3.430	6.969e-04	0.578	-1.072
BLM	-0.307	0.033	-3.399	7.778e-04	0.578	-1.146
MRPL46	-0.262	0.023	-3.371	8.551e-04	0.578	-1.209

The genes listed in the first column, are ten most relevant genes for discovering potential drug targets and biomarkers.

Similar genes and samples are clustered by the means of dendrograms which are formed in the heatmap.

5.5.3 Cox Ridge Model Fit and Validation

As mentioned in Sections 5.3.1 and 5.4, we apply the ridge regression and leave-one-out cross-validation to the data. The ridge method gives 11864 regression coefficients (for all 11864 features/genes). The optimal value of the tuning parameter λ obtained as $\lambda_{\text{opt}} = 22634.2$ and the cross-validated partial log-likelihood at the optimal λ is equal to $\text{CVPL}(\lambda_{\text{opt}}) = -723.258$. The cross-validated partial log-likelihood as a function of the penalty parameter λ is given in Figure 5.10 on the left. The function is unimodal and smooth with a peak at optimal value of λ. As the value of λ is positive and finite, we conclude that penalization has been helpful. The value of λ is quite large which is in line with the regression coefficients plotted in Figure 5.10 on the right hand side.

The predictive value of the model can now be assessed by looking at the histogram of the ridge PI, shown in Figure 5.11, and judging the possibility to discriminate between

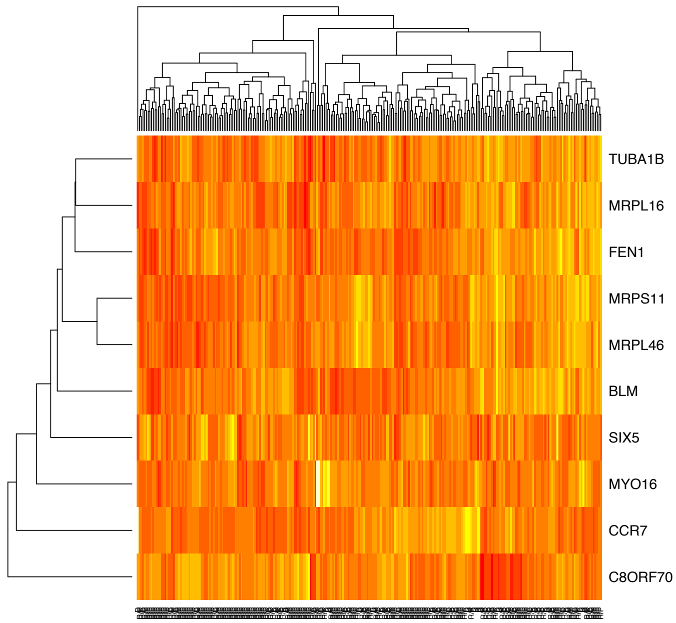


Figure 5.9: Heatmap for the top ten genes

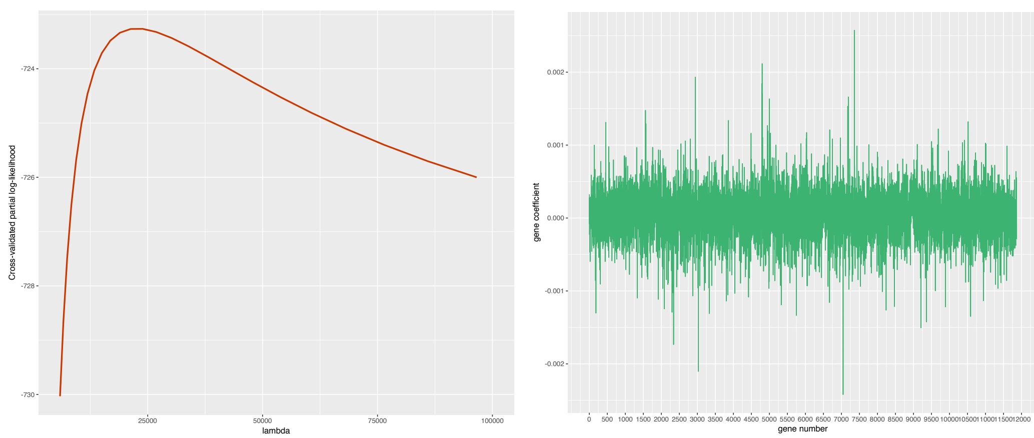


Figure 5.10: CVPL for penalized ridge regression (left) and Sequential plot of regression coefficients (right)

the patients with poor and good prognosis. One can read a detailed discussion one this in Van Houwelingen and Le Cessie (1990).

In a proportional hazards model, $h_i(t) = c_i h_0(t)$, the hazard ratio can be thought of as relative risk if the probability of failure is small, see (van Houwelingen and Putter, 2011, Chapter 2), which reflects on how the survivals are different. The leave-one-out relative

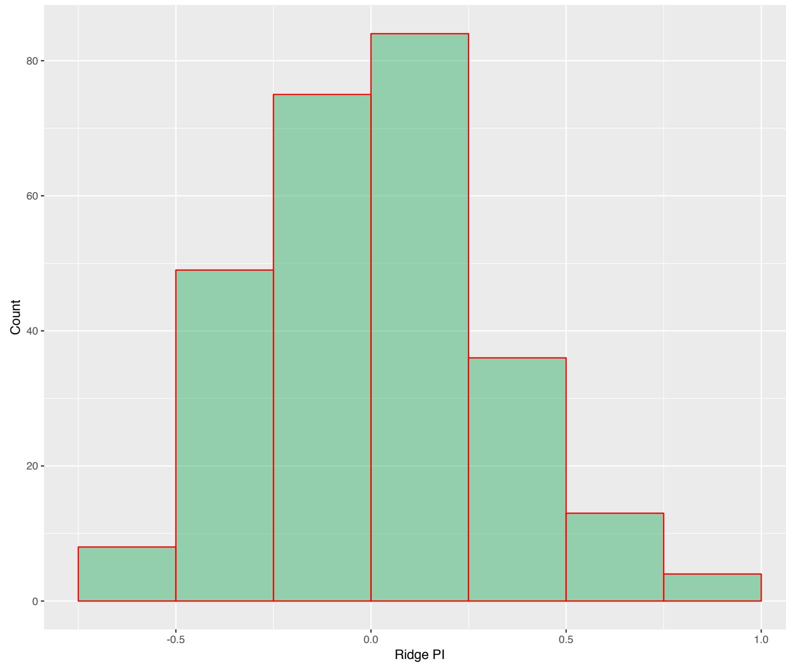


Figure 5.11: Histogram of the ridge prognostic index

risk when using the cross-validated partial log-likelihood can be obtained from

$$\mathrm{RR}_{\lambda,j}^{(-i)} = \exp\left(\boldsymbol{X}_j' \, \boldsymbol{\beta}_{\lambda}^{(-i)}\right), \tag{5.26}$$

which is actually the hazard ratio, using the notations defined in Section 5.4.1.

Here, the mean of the ridge PI is 0.008 and its SD is 0.303. Therefore, according to (5.26), on the PI scale, a difference of one standard deviation is associated with a hazard ratio of $\exp(0.303) = 1.354$.

The predicted survival curves for the 10th, 25th, 50th, 75th and 90th percentiles of the ridge PI are shown in Figure 5.12, from which the corresponding variation of PI in survival curves can be judged. This plot represents a model based prediction and therefore the validity of the proportional hazards assumption is crucial. Visually there is no clear evidence of overfitting or proportional hazards assumption violation. A more robust approach of checking this is discussed in Chapter 6.

5.6 Conclusions

In this chapter we discussed some methods for analyzing clinical data and also incorporating genetic data into survival models. A feature of genetic marker data is that there are typically many variables and so we have considered some approaches for dealing with high-dimensional data in survival models. In particular, we considered the use of regularization methods with the Cox proportional hazards model. Further approaches of analyzing clinical and genomic data, checking the model validity, and different ways of analyzing combined clinical and genomic data are discussed in Chapter 6. The application of this methodology to this particular dataset is a novel aspect of the work and allowed us to determine important genes and their predictive impact on survival.

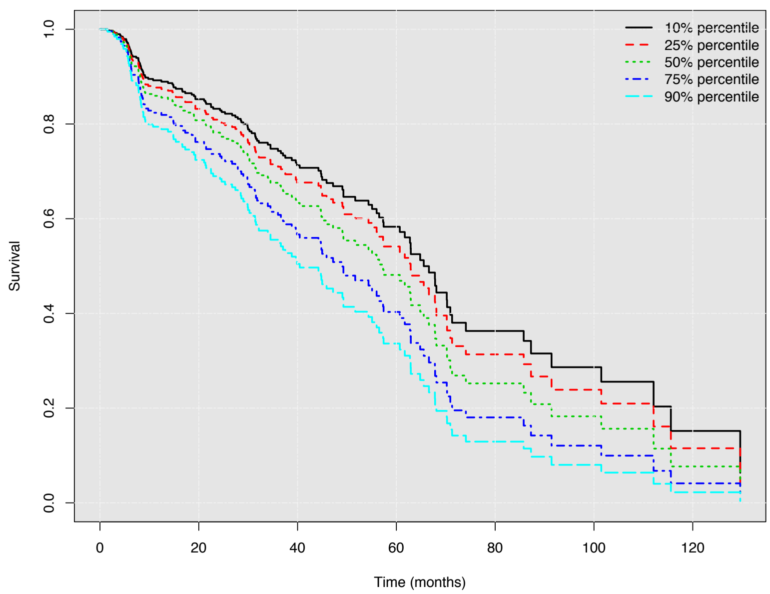


Figure 5.12: Estimated average survival curves for the 10th, 25th, 50th, 75th and 90th percentile of the ridge PI

Chapter 6

Discussion and Topics for Further Research

In this final chapter, we first give a brief overview of the other chapters of the thesis and their contribution. We then discuss further possible related research topics that could be undertaken.

6.1 Overview

In this thesis we have considered various aspects of survival/time to-event modeling. From some consideration of possible sampling schemes involving progressive censoring, through modeling including the use of finite mixtures and random effect models, to the incorporation of genetic biomarkers as problems with high-dimensional explanatory variables.

In Chapter 3 we investigated the properties of (progressive) Type-II censored data when they arise from a heterogeneous population and also explored its features when the component memberships are known. Moreover, we considered the case of mixture data without any kind of censoring. An extensive simulation study was conducted to compare the results over different scenarios. As the main focus of this chapter is on censored mixture data from heterogeneous populations, we have developed an `R` package `pcensmix` to conduct another simulation study specifically for exploring the results changing the component means and standard deviations, and the impact of how well-separated the components are. Also, we studied the changes in results when the undertaken censoring scheme is progressively applied compared to censoring all at the end (Type-II censoring). We concluded that under Type-II censoring, we have poorer estimates for parameters from the component with bigger mean with large SE's. Progressive Type-II censoring seems to protect against heterogeneity and thus better estimate the parameters from both the components, however, the resulting test time is longer under Progressive Type-II censoring, though shorter than the no censoring case. As r, the number of complete failures that defines the Type-II censoring stopping rule, becomes larger, the SE's become smaller, however, the experiment would take longer.

In Chapter 4, we considered another type of survival data: longitudinal clustered discrete survival data where censoring may occur due to failures that might happen after the final measurement time point. We applied generalized mixed effect models to the real data "insect" to model the mortality of insects with and without considering a time effect as a covariate. The "insect" dataset is a biological control assay collected with the purpose of studying the pathogenicity and virulence of a specific type of fungus, *Beauveria bassiana*. Different standard link functions in `R` are used and a new link function also is created for the sake of comparison between the models. Other models considered as a fitting choice include non-parametric models with and without time-trends. Models are compared in terms of their goodness of fit.

Chapter 5 is on analyzing methods for high-dimensional survival data and applications of them to a TCGA dataset from The Cancer Genome Atlas on ovarian carcinoma. The dataset clinically annotated as "HGS-OvCa" includes both clinical and genomic gene expression profile of patients which was measured with the motivation of increasing the successful treatment strategies in 2011.

Of course, in each of the areas that we have considered, there are further aspects that could be explored along with other possible extensions and applications. We present some possible avenues for future work in the following sections.

6.2 Joint Censoring in Mixture Models

To determine whether a procedure is more effective than an another alternative, we need to perform comparative studies. Comparative life testing is commonly used in industrial settings to compare different machines and components and also in medical settings to compare different diagnostic groups, treatment and control groups, etc. In comparative life-tests of two identical products manufactured on two different lines under the same conditions, joint censoring is of significant importance for some purposes such as estimating the mean lifetime of those products. First we look at two types of joint censoring.

6.2.1 Joint Type-II censoring

Suppose $X_1, \ldots, X_m$, the lifetimes of m specimens of product A, are independent identically distributed (i.i.d.) random variables from distribution function $F^{(1)}(x)$ and density function $f^{(1)}(x)$, and $Y_1, \ldots, Y_n$, the lifetimes of n specimens of product B, are i.i.d. random variables from distribution function $F^{(2)}(y)$ and density function $f^{(2)}(y)$. Further, suppose $Z_1 \leq \ldots \leq Z_N$, denote the order statistics of the $N = m + n$ random variables $\{X_1, \ldots, X_m; Y_1, \ldots, Y_n\}$. Then, under the joint Type-II censoring scheme, the observable data consist of (P, Z), where $Z = (Z_1, \ldots, Z_r)$, with r $(1 \leq r \leq N)$ being a pre-fixed integer, and $P = (P_1, \ldots, P_r)$ with P_i =1 or 0 according as Z_i is from an X-or$Y$-failure (population indicator for the failures). The likelihood is given by Balakrishnan and Rasouli (2008) as

$$\begin{aligned} L(\Theta|\boldsymbol{P}, \boldsymbol{Z}) &= \frac{m!\ n!}{(m-m_r)!\ (n-n_r)!} \prod_{i=1}^{r} \left[\{f^{(1)}(z_i)\}^{P_i} \{f^{(2)}(z_i)\}^{1-P_i} \right] \{\overline{F}^{(1)}(z_i)\}^{m-m_r} \\ &\quad \times \{\overline{F}^{(2)}(z_i)\}^{n-n_r}, \end{aligned}$$

where Θ is the parameter space and $\overline{F}^{(1)} = 1 - F^{(1)}$, $\overline{F}^{(2)} = 1 - F^{(2)}$ are the respective survival functions. Also $M_r = \sum_{i=1}^{r} P_i$ and $N_r = \sum_{i=1}^{r} (1 - P_i)$ denote the number of X- and Y-failures in Z.

6.2.2 Joint progressive Type-II censoring

Suppose that $\mathbf{X} = (X_1, \ldots, X_m)$ and $\mathbf{Y} = (Y_1, \ldots, Y_n)$ are i.i.d. random variables with the same distributional properties as defined in description of joint Type-II censoring. Also let $Z_1 \leq \ldots \leq Z_N$, $N = m + n$, denote the order statistics of the combined set $\{\mathbf{X}, \mathbf{Y}\}$.

Under JPTII censoring, at the time of the first failure Z_1 (that may be from either X or Y), R_1 units are randomly withdrawn from the remaining $N - 1$ surviving units. This process is continued at each subsequent failure time with R_i removed at the i-th observed failure. Here, the total number of observed failures r is pre-fixed and for the removals we can write $R_i = S_i + T_i$, $i = 1, 2, \ldots, r$, where S_i and T_i are the number of units withdrawn from the X and Y samples respectively.

The observed data can be summarized as $(\mathbf{P}, \mathbf{Z}, \mathbf{S})$, where P_i is a binary variable indicating which population the observation is from ($P_i = 1$ if Z_i is from $\mathbf{X}$). Figure 6.1 shows a schematic picture of this censoring scheme.

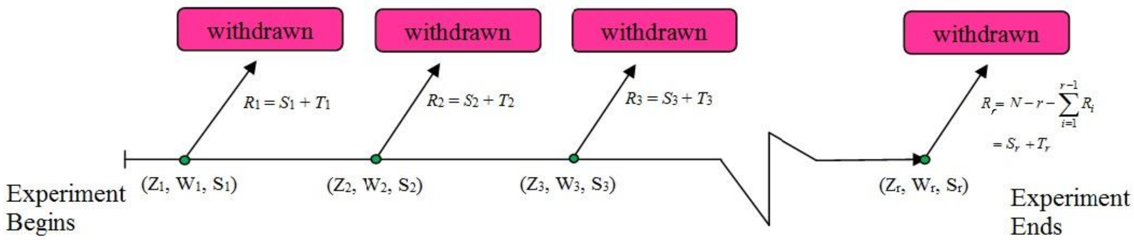


Figure 6.1: Schematic picture of JPTII censoring.

The likelihood function for component parameters Θ given the observed data is

$$L(\Theta|\boldsymbol{P}, \boldsymbol{z}, \boldsymbol{s}) = C \prod_{i=1}^{r} \left[\{f^{(1)}(z_i)\}^{P_i} \{f^{(2)}(z_i)\}^{1-P_i} \right] \{\overline{F}^{(1)}(z_i)\}^{s_i} \{\overline{F}^{(2)}(z_i)\}^{R_i - s_i},$$

where $\overline{F}^{(1)} = 1 - F^{(1)}$, $\overline{F}^{(2)} = 1 - F^{(2)}$ are the survivor functions and C is a normalising constant that depends only on the observed data.

For exponentially distributed lifetimes with $X \sim \text{Exponential}(\theta_1)$ and $Y \sim \text{Exponential}(\theta_2)$ the MLEs of θ_1 and θ_2 and the approximate SEs of parameters have been derived by Rasouli and Balakrishnan (2010) as follows:

$$\begin{aligned} \hat{\theta}_1 &= \frac{\sum_{i=1}^{r} P_i z_i + \sum_{i=1}^{r} z_i s_i}{\sum_{i=1}^{r} P_i} \\ \hat{\theta}_2 &= \frac{\sum_{i=1}^{r} (1 - P_i) w_i + \sum_{i=1}^{r} z_i (R_i - s_i)}{r - \sum_{i=1}^{r} P_i} \end{aligned}$$

$$\begin{aligned} \text{SE}(\hat{\theta}_1) &= \frac{\sum_{i=1}^{r} P_i z_i + \sum_{i=1}^{r} z_i s_i}{\sqrt{\{\sum_{i=1}^{r} P_i\}^3}} \\ \text{SE}(\hat{\theta}_2) &= \frac{\sum_{i=1}^{r} (1 - P_i) z_i + \sum_{i=1}^{r} z_i (R_i - s_i)}{\sqrt{\{r - \sum_{i=1}^{r} P_i\}^3}} \end{aligned}$$

This can be extended to two heterogeneous populations applying Type-II or progressive Type-II censoring over the two mixture populations, referred to as joint (progressive) Type-II censoring schemes. Generally speaking, under joint progressive censoring, the sample is based on two baseline samples $X_1, \ldots, X_{n_1}$ (product/Type A) and $Y_1, \ldots, Y_{n_2}$ (product/Type B) of independent random variables, where the X-sample follows F_1 distribution and Y-sample follows F_2 distribution and the progressive censoring is applied over the pooled sample, with the assumption that the type of failures and the type of censored individuals are known.

Rasouli and Balakrishnan (2010) considered such homogeneous joint populations for inference under joint progressive Type-II censoring when F_1 and F_2 are both exponential. However, it would be possible to extend this to include heterogeneous (mixed) populations for the samples.

6.3 Bayesian Generalized Linear Mixed Models

Generalized linear mixed models popularity have been growing in the last decades due to their ability to model different data types and being capable of handling multiple levels of dependency. The approaches we have taken in Chapter 4 are frequentist parametric and non-parametric over random effects. McCulloch and Neuhaus (2001) describes that the frequentist approach to the problem of taking the uncertainty of variance components into account is prevented by largely intractable distribution theory. In the Bayesian approach, this can be handled implicitly in the posterior distributions of parameters of interest. However, note that the Bayesian approach involves specification of the prior distributions of parameters which need to be chosen in some appropriate way.

Furthermore, the Bayesian approach might be computationally simpler to obtain variance estimates of the predictions of the random effects. As discussed in Booth and Hobert (1998), in a frequentist framework, second order estimation of the conditional standard error of prediction for the random effects requires bootstrapping fixed effects' and variance components' maximum likelihood estimates, while in the Bayesian framework, the main focus is on the posterior variance of the random effects given the data, which is a by-product of the MCMC output. Zhao et al. (2006) proposed a very general Bayesian generalized linear mixed effect model which allows for random effect for longitudinal data, crossed random effects, additive models with interaction, varying coefficient models, etc, using `WinBUGS` package.

As partly mentioned in Chapter 4, the packages in `R` for fitting a Bayesian generalized linear mixed includes `blme`, `MCMCglmm`, `brms`, `glmmML` and `glmmBUGS` which brings the BUGS language into `R`.

6.4 Methods for Clinical and Genomic Data Analysis

One of the typical standard ways of checking a predictive model is showing the relation between the survival outcome and the predictor by plotting the model based survival functions for different values of the predictor over its range, a plot like Figure 5.5. The more separate the curves, the more predictable the survivals are. A next step to this, is measuring the strength of that relation. Therefore, a *discriminative ability measurement* for a predictive survival model is used as a measure of goodness of fit, similar to R^2 in linear regression. For Binary outcome data, such as data from diagnosis tests, a popular method in clinical epidemiology to quantify how accurate the diagnostic tests are, is the plot of *Receiver Operation Characteristic* also known as ROC curve, which compares the distribution of the (ordinal) predictor in the groups of outcome levels. It basically plots the *True Positive Rate* versus the *False Positive Rate*. If the two underlying distributions are completely separated, it implies that the test perfectly discriminates. The area under the curve (AUC) interprets the measure of classification between the diseased and healthy patients.

In microarray gene expression studies, due to cost and lack of sample availability the number of replicated microarrays is usually small which could lead to inaccurate statistical hypothesis tests. It is worth mentioning that the Rosetta error model could be used as a pre-screening tool to reduce the number of genes to work on by setting a threshold on genes' p-values and keep the most significant ones based on the prefixed threshold. For more information on Rosetta error model, one could see van Houwelingen et al. (2006) or for extensive discussion see Weng et al. (2006).

In Chapter 5, we have plotted the predicted survival curves to show the relation between the predictor and the survival of patients. However, it might give a little too optimistic assessment of the predictive value of the model as the predictive value of the model is assessed in the data that it was derived from. One approach to correct the possible overfitting, is cross-validation. Cross-validated Cox model yields leave-one-out estimates of the regression coefficients $\hat{\beta}^{(-i)}$ and also the leave-one-out baseline hazard estimates $h_0^{(-i)}(t)$, all of which allow the calculation of predictive survival probabilities

$$\hat{S}^{(-i)}(t|x) = \exp\left(-\hat{H}_0^{(-i)}(t).\exp\left(x'\hat{\beta}^{(-i)}\right)\right),$$

which is the cross-validated version of (5.25) and can be used to generate a plot like Figure 5.5 for the prediction purpose. As a visual model checking, the cross-validated PI obtained in this way can be split to relatively equal size groups and used to compare the estimated survival curve in each group with the Kaplan-Meier plot.

The validity of proportional hazards assumption in model based predictions for example as in Figure 5.5 is crucial. A robust approach to check the validity taken in van Houwelingen and Putter (2011) is to use the cross-validated version of the PI explained above as a single time-varying covariate to refit a Cox model which seems to improve the predictive performance of the model. Note that this approach can be applied to both clinical and genomic data PIs.

Multiple different approaches are introduced in the literature for integrated analysis of clinical and genomic data. Bøvelstad et al. (2009) introduced the proportional hazard function as

$$h(t|X,z) = h_0(t)\exp\left(X'\beta + Z'\gamma\right),$$

containing both genomic and clinical covariates X and Z. This needs regularization on the genomic part handled by a ridge or lasso penalization on the β. As mentioned in van Houwelingen and Putter (2011), this method has a disadvantage of a sudden change in both β and γ due to correlation between clinical and genomic covariates. Instead, they introduced the idea of *Super model* with the Cox proportional hazards model given by

$$h(t|PI) = h_0(t)\exp\left(\alpha_1 \mathrm{PI}_{\mathrm{clin}}(Z) + \alpha_2 \mathrm{PI}_{\mathrm{gen}}(X)\right),$$

where $\mathrm{PI}_{\mathrm{clin}}$ and $\mathrm{PI}_{\mathrm{gen}}(X)$ indicate the prognostic indexes from two separate Cox models defined based on clinical and genomic data respectively. Further, they combine this with the idea of *landmark prediction*. The reader is referred to (van Houwelingen and Putter, 2011, Section 12.2) for more detailed information.

Appendix: R package “`pcensmix`”

The current version of this package on CRAN provides functions to generate two-component mixture data from various different mixture distributions, generate progressively Type-II censored data in a mixture structure and fit a normal mixture model to right censored data using a constrained EM algorithm.

It can also create a progressively Type-II censored version of a given real dataset and fit a normal mixture model to.

The help file document describing the functions in the package is followed in the next page.

Package 'pcensmix'

July 24, 2017

Type Package

Title Model Fitting to Progressively Censored Mixture Data

Version 1.2-1

Depends R (>= 3.3.3), stats

Imports utils

Description Functions for generating progressively Type-II censored data in a mixture structure and fitting models using a constrained EM algorithm. It can also create a progressive Type-II censored version of a given real dataset to be considered for model fitting.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Lida Fallah [aut, cre],
John Hinde [aut]

Maintainer Lida Fallah <l.fallah22@gmail.com>

Repository CRAN

Date/Publication 2017-07-24 19:01:10 UTC

R topics documented:

`pcensmix-package` *Generate and fit a model to censored mixture data*

Description

This package provides functions to generate two-component mixture data from various different mixture distribution, generate progressive Type-II censored data in a mixture structure and fit a normal mixture model using a constrained EM algorithm. In addition, it can create a progressive Type-II censored version of a given real dataset and fit a normal mixture model to. Main functions are pcgen, pcensmixSim and pcensmixR. Example datasets are included for illustration.

Details

Package: pcensmix

Type: Package

Version: 1.2-1

Date: 2017-07-24

License: GPL (>= 2)

Author(s)

Lida Fallah <l.fallah22@gmail.com> and John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

References

Aitkin, M., Francis, B., Hinde, J. and Darnell, R., (2009). Statistical Modelling in R. Oxford: Oxford University Press.

Balakrishnan, N. and Aggarwala, R., (2000). Progressive Censoring: Theory, Methods, and Applications. Springer Science & Business Media.

Hathaway, R.J., (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. The Annals of Statistics, 795-800.

McLachlan, G. and Krishnan, T., (2007). The EM Algorithm and Extensions. John Wiley & Sons.

McLachlan, G. and Peel, D., (2004). Finite Mixture Models. John Wiley & Sons.

`blood` *Blood pressure of mine workers.*

Description

A dataframe containing the blood pressure of a population of mine workers in Ghana.

Usage

```
blood
```

Format

A data frame with 495 rows and 3 variables

- Systolic.BP. Systolic blood pressure
- Diastolic.BP. Diastolic blood pressure
- Amplitude. Amplitude

References

Boehning, D., (2000). Computer-assisted Analysis of Mixtures and Applications: Meta-analysis, Disease mapping and Others. CRC press.

Gunga, H.C., Forson, K., Amegby, N. and Kirsch, K., (1991). Lebensbedingungen und gesundheitszustand von berg-und fabrikarbeitern im tropischen regenwald von Ghana. Arbeitsmedizin Sozialmedizin Praventivmediz in, 17-25.

`insulate` *Failure times of insulating fluid.*

Description

A dataset containing measurements of 19 failure times (in minutes) for an insulating fluid between two electrodes subject to a voltage of 34 KV, see Nelson (1982).

Usage

```
insulate
```

Format

A vector of length 19.

References

Nelson, W., (1982). Applied Life Data Analysis. Wiley, New York.

`mixgen` *Generating Mixture Datasets*

Description

This function generates two-component mixture data from a various different mixture distributions.

Usage

```
mixgen(N, dist1, dist2, control)
```

Arguments

`N`	population size.
`dist1, dist2`	respective distributions of the first and second mixture components to be simulated from. For Normal, Log-normal, Weibull, Gamma, Cauchy and Beta distributions, they must be provided as `'norm'`, `'lnorm'`, `'weibull'`, `'gamma'`, `'cauchy'` and `'beta'` respectively. The Exponential distribution can be included as a Gamma distribution with scale parameter one.
`control`	a list of parameters for controlling the simulation process. This includes parameters of the first and second mixture component distributions and the mixture proportion π which should be provided in the order mentioned, i.e., parameters for the first component come first, those of the second component come after and then the value of the mixing proportion. All values should be provided in a list of numerics. Note parameters for each component distribution should be added in the order as required in their generator functions, see `rnorm`, `rlnorm`, `rweibull`, `rgamma`, `rcauchy` and `rbeta`.

Details

It generates a two-component mixture dataset from a density function

$$\pi f_1 + (1 - \pi) f_2,$$

where f_1 and f_2 are the first and the second mixture component distributions respectively.

Value

An object of class `data.frame` containing the following information:

`z`	mixture data
`label`	component indicator

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

pcgen

Examples

```
## Generate a sample from a two component Normal-Weibull mixture distribution
## with mixing components as N(12, 2) and Weibull(15, 4), mixing proportion 0.3
## and size of N = 20.

mixture<- mixgen(N = 20, dist1 = 'norm', dist2 = 'weibull', control = list(12, 2, 15, 4, 0.3))
```

---

pcensmixR *Fitting a Normal Mixture Model to a Real Progressive Type-II Censored Mixture Data Using EM Algorithm*

Description

This function uses a two-layer EM algorithm to fit a mixture model to progressive Type-II censored mixture data by estimating the latent mixture components and the censored data.

Usage

```
pcensmixR(Pdat, ...)

## S3 method for class 'pcgen'
pcensmixR(Pdat, start, iteration = 1e+05, INERiter = 20,
  warn = FALSE, ...)
```

Arguments

`Pdat`	an object of class `"pcgen"` created by function `pcgen` or a two-column matrix (or data.frame) with first column giving a vector of censored version of a two-component mixed normal data, and the other one indicating the censoring status associated with them (1 if not censored, otherwise zero).
`...`	additinal arguments to pass by.
`start`	a numeric vector; used as starting values for the EM algorithm.
`iteration`	the maximum number of required iteration for the EM algorithm until convergence– default value is 1e+05.
`INERiter`	the maximum number of required iteration for the second EM algorithm– default is 20.
`warn`	logical. shows warning messages if `TRUE`, if there is any– default is `FALSE`.

Details

This function fits a two-component normal mixture model to a given progressive Type-II censored data.

It uses a two-layer EM algorithm for fitting the model. Generally speaking, the first layer estimates the mixture component latent variables, in the E-step, by finding their conditional expected values given the current parameter estimates and the data; and the second layer consists of another EM algorithm to estimate the missing censored data and eventually the parameters of interest. The layers are repeated until convergence achieved.

Value

`pcensmixR` gives an object of class `data.frame` containing the following components:

`muhat1,sigmahat1`
: component one parameter estimates ($\hat{\mu}_1$, $\hat{\sigma}_1$)

`muhat2,sigmahat2`
: component two parameter estimates ($\hat{\mu}_2$, $\hat{\sigma}_2$)

`pihat`
: estimation of mixture proportion $\hat{\pi}$

`se.muhat1,se.sigmahat1`
: standard errors of $\hat{\mu}_1$ and $\hat{\sigma}_1$

`se.muhat2,se.sigmahat2`
: standard errors of $\hat{\mu}_2$ and $\hat{\sigma}_2$

`se.pihat`
: standard error of $\hat{\pi}$

`no.fails.comp1,no.fails.comp2`
: number of failures from each mixture component

`no.cens.comp1,no.cens.comp2`
: number of censored observations from each mixture component

`ll`
: log-likelihood value

Note

See `pcgen` for the definition of censored version of data.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

`pcgen`, `pcensmixSim`

Examples

```
## Example 1: fit a mixture model to 'insulate' data
set.seed(107)
Pdat<- pcgen(r = 15, p = 0.6, data = insulate)
pcensmixR(Pdat, start = c(5, 3, 35, 20, 0.6))

## Not run:
## Example 2: fit a mixture model to 'Systolic blood pressure' data
set.seed(1010)
pcensmixR(Pdat = pcgen(360, 0.35, blood$Systolic.BP),
              start = c(120, 15, 150, 20, 0.6))
## End(Not run)
```

`pcensmixSim` *Fitting a Normal Mixture Model to a Simulated Progressive Type-II Censored Data Using EM Algorithm*

Description

This function fits a normal mixture model to progressive Type-II censored mixture data by dealing with the two aspects of missing data, latent mixture components and the censored data, using a maximum likelihood estimation through a constrained two-layer EM algorithm.

Usage

```
pcensmixSim(Pdat, ...)

## S3 method for class 'pcgen'
pcensmixSim(Pdat, r, p, param, iteration = 1e+05,
  INERiter = 20, ...)
```

Arguments

`Pdat`	an object of class `"pcgen"` created by function `pcgen` or a two-column matrix (or data.frame) with first column giving a vector of censored version of a two-component mixed normal data, and the other one indicating the censoring status associated with them (1 if not censored, otherwise zero).
`...`	additinal arguments to pass by.
`r`	desired number of failures to observe.
`p`	a parameter controlling the amount of censoring. The action of censoring individuals after each failure occurs with probabilty `p` from binomial distribution at each stage. If `p = 0`, there will be no censoring.
`param`	a numeric vector; used as starting values for the EM and simulating a new data to replace in case of happening singularity in the likelihood.

`iteration` the maximum number of required iteration for the EM algorithm until convergence– default value is 1e+05.

`INERiter` the maximum number of required iteration for the second EM algorithm– default is 20.

Details

This function fits a two-component normal mixture model to simulated progressive Type-II censored data with density function

$$\pi(\frac{1}{\sigma_1})\,\phi\Big[\frac{(z-\mu_1)}{\sigma_1}\Big] + (1-\pi)(\frac{1}{\sigma_2})\,\phi\Big[\frac{(z-\mu_2)}{\sigma_2}\Big]$$

where ϕ is the standard normal density.

It uses a constrained two-layer EM algorithm to deal with the two forms of missing data: the censored survival times and the mixture component labels. Given the EM algorithm is at a particular iteration: (i) first, in the E-step it obtains the mixture component indicator estimates given the current parameter estimates and the observed data. (ii) Next, for re-estimation of the unknown parameters, a new EM algorithm is nested in the M-step of the initial EM algorithm to deal with the estimation of the missing censored survival times and consequently building the maximum likelihood equations. These steps are repeated until the model converges.

Value

`pcensmixSim` gives an object of class `data.frame` containing the following components:

`muhat1,sigmahat1`
component one parameter estimates ($\hat{\mu}_1$, $\hat{\sigma}_1$)

`muhat2,sigmahat2`
component two parameter estimates ($\hat{\mu}_2$, $\hat{\sigma_2}$

`pihat` estimation of mixture proportion $\hat{\pi}$

`se.muhat1,se.sigmahat1`
standard errors of $\hat{\mu}_1$ and $\hat{\sigma_1}$

`se.muhat2,se.sigmahat2`
standard errors of $\hat{\mu}_2$ and $\hat{\sigma}_2$

`se.pihat` standard error of $\hat{\pi}$

`no.fails.comp1,no.fails.comp2`
number of failures from each mixture component

`no.cens.comp1,no.cens.comp2`
number of censored observations from each mixture component

`ll` log-likelihood value

`datachange_flag`
`TRUE` if data has been replaced by a newly generated one

Note

- In fitting the model, to overcome the problem of singularity and model non-identifiability that might happen in some cases depending on the true means and standard deviations of the components, we use the constraint proposed by Hathaway (1985). Based on this, the ratios of the standard deviations are considered to be greater than a pre-fixed constant. We consider the constraints $\sigma_1/\sigma_2 > 0.1$ and $\sigma_2/\sigma_1 > 0.1$ which lead to obtain a parameter space with no singularities. In case this conditions doesn't hold, the data will be replaced by a new simulated one and `datachange_flag` will appear as `TRUE` in the output.
- See `pcgen` for the definition of censored version of data.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

`pcgen`, `run_pcensmix`

Examples

```
## Not run:
set.seed(100)

Pdat<- pcgen(r = 60, p = 0.3, data = mixgen(N = 100, dist1 = 'norm',
                 dist2 = 'norm', control = list(12, 2, 14, 5, 0.35)))
pcensmixSim(Pdat, r = 60, p = 0.3, param=c(12, 2, 14, 5, 0.35))
## End(Not run)
```

`pcgen` *Creating a Progressively Type-II Censored Version of a Given Dataset*

Description

This function implements an algorithm for generating a progressive Type-II censored version of a specified dataset.

Usage

```
pcgen(r, p, data)
```

Arguments

`r`	desired number of failures to observe.
`p`	a parameter controlling the amount of censoring. The action of censoring individuals after each failure occurs with probabilty `p` from binomial distribution at each stage. If `p = 0`, no censoring will happen.
`data`	a numeric vector of a real dataset (mixture/not mixture) or an object of class `data.frame` generated by `mixgen`.

Details

It creates a progressive Type-II censored version of a given real dataset or a simulated dataset from `mixgen`. The output of this function can be passed as an argument to `pcensmixR` or `pcensmixSim` for the purpose of fitting a normal mixture model to the progressively censored dataset.

Value

An object of class `"pcgen"` containing the following information:

`original_data`	original mixture data
`label`	component membership indicator for the original simulated mixture data. This will not be returned if a real data has been used.
`censored_version_of_data`	progressive Type-II censored version of data, i.e., each observation is equal to the actual observed survival time in the event of failure and is equal to the latest observe failure time if it is associated to an unobserved censored observation. Notice that they are order statistics.
`component_indicator`	component indicator associated with the censored_verison_of_data. This will not be returned if a real data has been used.
`censoring_indicator`	censoring indicator associated with the censored_verison_of_data.

Note

See `print.pcgen` for printing data of class `"pcgen"`.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

`mixgen`, `print.pcgen`.

Examples

```
## 1. Generate a progressive Type-II censored data from a simulated mixture data with
## allowing for censoring with controlling parameters p = 0.3 and r = 12.
set.seed(0)
mixture <- mixgen(N = 20, dist1 = 'norm', dist2 = 'weibull', control = list(12, 2, 15, 4, 0.3))
Pdat0 <- pcgen(r = 12, p = 0.3, data = mixture)
print(Pdat0)


## 2. Examples of generating a progresively Type-II censored data

set.seed(0)
Pdat1 <- pcgen(r = 6, p = 0.3, data = insulate)
print(Pdat1)

set.seed(100)
Pdat2 <- pcgen(r = 260, p = 0.35, data = blood$Systolic.BP)
print(Pdat2)
```

`print.pcgen` *Print Method for pcgen Objects*

Description

This function prints the progressive censored data generated by the S3 class `pcgen`.

Usage

```
## S3 method for class 'pcgen'
print(x, ...)
```

Arguments

`x`	object of class pcgen.
`...`	optional arguments to pass by.

Value

This function uses the generic function `print` to print the dataset of class `"pcgen"` in a nice format.

Examples

```
## Generate a two component normal mixture data,
Pdat <- pcgen(r = 80, p = 0.3, data = mixgen(N = 100, dist1 = 'norm',
              dist2 = 'norm', control = list(12, 2, 14, 4, 0.3)))
# and print it.
print(Pdat)
```

`run_pcensmix` *Generating Progressively Type-II Censored Mixture Data and Fitting a Model*

Description

This function implements an algorithm using the `mixgen`, `pcgen` and `pcensmixSim` functions to generate data and fit a model using EM algorithm with a specified number of iterations.

Usage

```
run_pcensmix(N, r, p, param, repetition = 100)
```

Arguments

`N`	population size.
`r`	required number of failures to observe.
`p`	a parameter controlling the amount of censoring. The action of censoring individuals after each failure occurs with probabilty `p` from a binomial distribution at each stage. If `p = 0`, there will be no censoring.
`param`	a numeric vector; used as starting values for the EM and simulating a new data to replace in case of happening singularity in the likelihood.
`repetition`	the required number of repetition of the algorithm– default is 100.

Value

It returns the parameter estimates given by `pcensmixSim` with the desired number of repetitions. In each repetition it generates a new normal mixture progressive Type-II censored dataset from the same true parameter values and fits a model.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

`pcgen`, `pcensmixSim`, `mixgen`

Examples

```
## Not run:

## Example 1: with very well separated mixture components
set.seed(3)
f1 <- run_pcensmix(N = 160, r = 120, p = 0.3, param = c(10, 2, 25, 4, 0.3), repetition = 100)
colMeans(f1)
```

Bibliography

Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing*, **6**(3), 251–262. [89, 117]

Aitkin, M. A., Francis, B., Hinde, J., and Darnell, R. (2009). *Statistical Modelling in R*. Oxford University Press Oxford. [33, 34, 47]

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**(6), 716–723. [152]

Almeida, J. d., Alves, S., and Pereira, R. (1997). Selection of beauveria spp. isolates for control of the termite heterotermes tenuis (hagen, 1858). *Journal of Applied Entomology*, **121**(1-5), 539–543. [89]

Aranda-Ordaz, F. J. (1981). On two families of transformations to additivity for binary response data. *Biometrika*, pages 357–363. [94]

Ateya, S. F. (2014). Maximum likelihood estimation under a finite mixture of generalized exponential distributions based on censored data. *Statistical Papers*, **55**, 311–325. [27]

Baker, S. G. (1992). A simple method for computing the observed information matrix when using the em algorithm with categorical data. *Journal of Computational and Graphical Statistics*, **1**(1), 63–76. [17]

Balakrishnan, N. (2007). Progressive censoring methodology: an appraisal. *Test*, **16**(2), 211–259. [11]

Balakrishnan, N. and Aggarwala, R. (2000). *Progressive censoring: theory, methods, and applications*. Springer Science & Business Media. [7, 11]

Balakrishnan, N. and Cramer, E. (2014). The art of progressive censoring. *Birkhauser, New York*. [10, 11, 18]

Balakrishnan, N. and Rasouli, A. (2008). Exact likelihood inference for two exponential populations under joint Type-II censoring. *Computational Statistics and Data Analysis*, **52**(5), 2725–2738. [167]

Bates, D., Mchler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, Articles*, **67**(1), 1–48. [108]

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, pages 289–300. [161]

Bolker, B. et al. (2017). Glmm faq. `https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html`. [108, 129]

Bolker, B. and Team, R. D. C. (2014). *bbmle: Tools for general maximum likelihood estimation*. R package version 1.0.17. [17]

Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., and White, J.-S. S. (2009). Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in ecology & evolution*, **24**(3), 127–135. [23]

Booth, J. G. and Hobert, J. P. (1998). Standard errors of prediction in generalized linear mixed models. *Journal of the American Statistical Association*, **93**(441), 262–272. [169]

Bøvelstad, H. M., Nygård, S., and Borgan, Ø. (2009). Survival prediction from clinico-genomic models-a comparative study. *BMC bioinformatics*, **10**(1), 413. [171]

Breslow, N. E. (1972). Discussion of professor coxs paper. *Journal of the Royal Statistical Society: Series B*, **34**, 216–217. [148]

Cancer Genome Atlas Research Network and others (2011). Integrated genomic analyses of ovarian carcinoma. *Nature*, **474**, 609–615. [25]

Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., and Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, **3**(1), 140. [137]

Clarke, M. and Oxman, A. (2000). *Cochrane reviewers' handbook*. Update Software. [136]

Cohen, A. C. (1963). Progressively censored samples in life testing. *Technometrics*, **5**(3), 327–339. [9]

Collett, D. (2015). *Modelling survival data in medical research*. CRC press. [147]

Cox, D. R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society: Series B*, **34**, 187–220. [26, 145, 147]

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, pages 1–38. [14, 34]

Dobson, A. J. and Barnett, A. (2008). *An Introduction to Generalized Linear odels*. CRC press. [22]

Efron, B. and Hastie, T. (2016). *Computer Age Statistical Inference*, volume 5. Cambridge University Press. [136]

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, **32**(2), 407–499. [150]

Einbeck, J. and Hinde, J. (2006). A note on NPML estimation for exponential family regression models with unspecified dispersion parameter. *Austrian journal of statistics*, **35**(2&3), 233–243. [89, 117]

Everitt, B. and Hand, D. (1981). *Finite Mixture Distributions*. Wiley Online Library. [27]

Fallah, L., Golparvar, L., and Parsian, A. (2016). Inference based on progressively censored sample from Pareto population. *Communications in Statistics-Theory and Methods*, **45**(1), 9–24. [11, 26]

Faraway, J. J. (2006). *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press. [22, 115]

Goeman, J. J., Oosting, J., Cleton-Jansen, A.-M., Anninga, J. K., and Van Houwelingen, H. C. (2005). Testing association of a pathway with survival using gene expression data. *Bioinformatics*, **21**(9), 1950–1957. [158]

Gui, J. and Li, H. (2005). Penalized cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. *Bioinformatics*, **21**(13), 3001–3008. [25]

Gupta, A. (1952). Estimation of the mean and standard deviation of a normal population from a censored sample. *Biometrika*, **39**, 260–273. [26]

Hastie, Trevor, T. R. and Friedman, J. (2009). *The Elements of Statistical Learning*, volume 1. [149, 152]

Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, **13**, 795–800. [60, 67]

Herd, G. R. (1956). Estimation of the parameters of a population from a multi-censored sample. [9]

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67. [149]

Hosmer Jr, D. W. and Lemeshow, S. (1999). Applied survival analysis: Regression modelling of time to event data (1999). [147]

Hwang, T., Sicotte, H., Tian, Z., Wu, B., Kocher, J.-P., Wigle, D. A., Kumar, V., and Kuang, R. (2008). Robust and efficient identification of biomarkers by classifying features on graphs. *Bioinformatics*, **24**(18), 2023–2029. [137]

Jung, S.-H., Owzar, K., and George, S. L. (2005). A multiple testing procedure to associate gene expression levels with survival. *Statistics in medicine*, **24**(20), 3077–3088. [25]

Kalbfleisch, J. D. and Prentice, R. L. (2011). *The statistical analysis of failure time data*, volume 360. John Wiley & Sons. [147]

Karlis, D. (2009). A note on the exponential poisson distribution: A nested EM algorithm. *Computational Statistics & Data Analysis*, **53**(4), 894–899. [16]

Koonings, P. P., Campbell, K., MISHELL Jr, D. R., and Grimes, D. A. (1989). Relative frequency of primary ovarian neoplasms: a 10-year review. *Obstetrics & Gynecology*, **74**(6), 921–926. [153]

Kuo, L. and Peng, F. (2000). A mixture-model approach to the analysis of survival data. *BIOSTATISTICS-BASEL-*, **5**, 255–271. [27, 46, 68]

Laird, N. (1978). Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, **73**(364), 805–811. [89, 117]

Lawless, J. F. (2011). *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons. [7, 13, 26]

Lee, Y., Nelder, J. A., and Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. CRC Press. [23]

Levine, R. A. and Casella, G. (2001). Implementations of the monte carlo EM algorithm. *Journal of Computational and Graphical Statistics*, **10**(3), 422–439. [15]

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, **24**(9), 1175–1182. [137]

Li, G. and Lin, C. (2009). Analysis of two-sample censored data using a semiparametric mixture model. *Acta Mathematicae Applicatae Sinica, English Series*, **25**, 389–398. [27]

Litière, S., Alonso, A., and Molenberghs, G. (2008). The impact of a misspecified random-effects distribution on the estimation and the performance of inferential procedures in generalized linear mixed models. *Statistics in Medicine*, **27**(16), 3125–3144. [89]

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B*, pages 226–233. [17, 18, 36]

McCulloch, C. E. and Neuhaus, J. M. (2001). *Generalized linear mixed models*. Wiley Online Library. [23, 169]

McCulloch, C. E. and Neuhaus, J. M. (2011). Misspecifying the shape of a random effects distribution: why getting it wrong may not matter. *Statistical Science*, pages 388–402. [89]

McCulloch, Charles E, N. J. M. and Searle, S. R. (2008). *Generalized, Linear, and Mixed Models*. John Wiley & Sons, 2nd edition. [112]

McLachlan, G. and Krishnan, T. (2007). *The EM algorithm and extensions*, volume 382. John Wiley & Sons. [17, 33]

McLachlan, G. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons. [13, 27]

Molenberghs, G. and Verbeke, G. (2004). An introduction to (generalized (non) linear mixed models. In *Explanatory Item Response Models*, pages 111–153. Springer. [23]

Nelder, J. and Baker, R. (1972). *Generalized linear models*. Wiley Online Library. [22]

Neuhaus, J. M., McCulloch, C. E., and Boylan, R. (2013). Estimation of covariate effects in generalized linear mixed models with a misspecified distribution ofrandom intercepts and slopes. *Statistics in Medicine*, **32**(14), 2419–2429. [89]

Ng, H., Chan, P., and Balakrishnan, N. (2002). Estimation of parameters from progressively censored data using EM algorithm. *Computational Statistics and Data Analysis*, **39**(4), 371–386. [18]

Oakes, D. (1999). Direct calculation of the information matrix via the EM. *Journal of the Royal Statistical Society: Series B*, **61**(2), 479–482. [17]

Park, C. and Lee, S. B. (2012). Parameter estimation from censored samples using the expectation-maximization algorithm. *arXiv preprint arXiv:1203.3880*. [26]

Park, P. J. (2005). Gene expression data and survival analysis. In *Methods of Microarray Data Analysis*, pages 21–34. Springer. [24]

Pipper, C. B., Ritz, C., and Scheike, T. (2011). Explained variation in a fully specified model for data-grouped survival data. *Biometrics*, **67**(4), 1361–1368. [88, 89]

Policello II, G. E. (1981). Conditional maximum likelihood estimation in gaussian mixtures. In Fagerberg, J., Mowery, D. C., and Nelson, R. R., editors, *Statistical Distributions in Scientific Work*. Springer. [62]

Popescu, C. A. and Wong, Y. S. (2005). Nested monte carlo EM algorithm for switching state-space models. *IEEE Transactions on Knowledge and Data Engineering*, **17**(12), 1653–1663. [16]

Powell, L. A. (2007). Approximating variance of demographic parameters using the delta method: a reference for avian biologists. *The Condor*, **109**(4), 949–954. [45]

Prinja, S., Gupta, N., Verma, R., et al. (2010). Censoring in clinical trials: Review of survival analysis techniques. *Indian Journal of Community Medicine*, **35**(2), 217. [136]

Putter, H., van der Hage, J., de Bock, G., Elgalta, R., and van de Velde, C. J. (2006). Estimation and prediction in a multi-state model for breast cancer. *Biometrical Journal*, **48**(3), 366–380. [142]

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. [69]

Rasouli, A. and Balakrishnan, N. (2010). Exact likelihood inference for two exponential populations under joint progressive Type-II censoring. *Communications in Statistics-Theory and Methods*, **39**(12), 2172–2191. [26, 67, 168, 169]

Rizopoulos, D., Verbeke, G., and Molenberghs, G. (2008). Shared parameter models under random effects misspecification. *Biometrika*, **95**(1), 63–74. [89]

Santos, C. A. and Achcar, J. A. (2011). A Bayesian analysis in the presence of covariates for multivariate survival data: An example of application. *Revista Colombiana de Estadística*, **34**, 111–131. [27]

Seidman, J. D., Horkayne-Szakaly, I., Haiba, M., Boice, C. R., Kurman, R. J., and Ronnett, B. M. (2004). The histologic type and stage distribution of ovarian carcinomas of surface epithelial origin. *International Journal of Gynecological Pathology*, **23**(1), 41–44. [153]

Shoemaker, J. S. and Lin, S. M. (2004). *Methods of Microarray Data Analysis IV*, volume 4. Springer Science & Business Media. [25]

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for coxs proportional hazards model via coordinate descent. *Journal of statistical software*, **39**(5), 1. [152]

Swinscow, T. D., Campbell, M. J., et al. (2002). *Statistics at square one*. Bmj London. [138]

Tanner, M. A. (1991). *Tools for statistical inference*, volume 3. Springer. [18, 27]

Teicher, H. (1963). Identifiability of finite mixtures. *The Annals of Mathematical Statistics*, pages 1265–1269. [13]

Tibshirani, R. J. and Ciampi, A. (1983). A family of proportional-and additive-hazards models for survival data. *Biometrics*, pages 141–147. [89, 92, 93]

Vaida, F. and Xu, R. (2000). Proportional hazards model with random effects. *Statistics in Medicine*, **19**(24), 3309–3324. [89]

Van Belle, G., Fisher, L. D., Heagerty, P. J., and Lumley, T. (2004). *Biostatistics: a methodology for the health sciences*, volume 519. John Wiley & Sons. [22]

Van De Vijver, M. J., He, Y. D., Van't Veer, L. J., Dai, H., Hart, A. A., Voskuil, D. W., Schreiber, G. J., Peterse, J. L., Roberts, C., Marton, M. J., et al. (2002). A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine*, **347**(25), 1999–2009. [25]

Van Dyk, D. A. (2000). Nesting EM algorithms for computational efficiency. *Statistica Sinica*, pages 203–225. [16]

van Houwelingen, Hans C, B. T. H. A. A., van't Veer, L. J., and Wessels, L. F. (2006). Cross-validated cox regression on microarray gene expression data. *Statistics in Medicine*, **25**, 3201–3216. [25, 148, 170]

van Houwelingen, H. and Putter, H. (2011). *Dynamic prediction in clinical survival analysis*. CRC Press. [25, 142, 158, 162, 171]

Van Houwelingen, J. and Le Cessie, S. (1990). Predictive value of statistical models. *Statistics in medicine*, **9**(11), 1303–1325. [162]

Verbeke, G. and Molenberghs, G. (2013). The gradient function as an exploratory goodness-of-fit assessment of the random-effects distribution in mixed models. *Biostatistics*, **14**(3), 477–490. [89]

Walters, S. J. (2012). Analyzing time to event outcomes with a cox regression model. *Wiley Interdisciplinary Reviews: Computational Statistics*, **4**(3), 310–315. [148]

Wang, X. V., Verhaak, R. G., Purdom, E., Spellman, P. T., and Speed, T. P. (2011). Unifying gene expression measures from multiple platforms using factor analysis. *PloS one*, **6**(3), e17691. [154]

Weng, L., Dai, H., Zhan, Y., He, Y., Stepaniants, S. B., and Bassett, D. E. (2006). Rosetta error model for gene expression analysis. *Bioinformatics*, **22**(9), 1111–1121. [170]

Whitfield, C. W., Cziko, A.-M., and Robinson, G. E. (2003). Gene expression profiles in the brain predict behavior in individual honey bees. *Science*, **302**(5643), 296–299. [24]

Wolynetz, M. (1979). Algorithm AS 139: Maximum likelihood estimation in a linear model from confined and censored normal data. *Applied Statistics*, pages 195–206. [36]

Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, **11**, 95–103. [53]

Zhang, W., Ota, T., Shridhar, V., Chien, J., Wu, B., and Kuang, R. (2013). Network-based survival analysis reveals subnetwork signatures for predicting outcomes of ovarian cancer treatment. *PLoS Computational Biology*, **9**(3), e1002975. [137, 148, 149]

Zhao, Y., Staudenmayer, J., Coull, B. A., and Wand, M. P. (2006). General design bayesian generalized linear mixed models. *Statistical Science*, pages 35–51. [169]