National Institute for Applied Statistics Research Australia

University of Wollongong, Australia

Working Paper

17-12

Modern Applications of Linear Mixed Models in Case Studies: Course Notes

Brian Cullis, Sue Welham, Beverley Gogel,
David Butler, and Robin Thompson

National Institute for Applied Statistics Research Australia, University of Wollongong,
Wollongong NSW 2522, Australia Phone +61 2 4221 5076, Fax +61 2 4221 4998.
Email: karink@uow.edu.au

THE UNIVERSITY of ADELAIDE

GRDC
Grains Research & Development Corporation

Modern Applications of Linear Mixed Models in Case Studies: Course Notes

Brian Cullis
University of Wollongong
email:bcullis@uow.edu.au

Sue Welham
VSN International
email: sue.welham@vsni.co.uk

Beverley Gogel
University of Adelaide
email: beverley.gogel@adelaide.edu.au

David Butler
Queensland Department of Primary Industries
email: david.butler@daff.qld.gov.au

Robin Thompson
Rothamsted Research
email: robin.thompson@rothamsted.ac.uk

November 30, 2012

Contents

Preliminaries

In this half-day course we give a brief overview of linear mixed models and their specification and fitting by the asreml function implemented in R.

We do not provide any details of theory on inference in linear mixed models, except where directly relevant, as this can be found elsewhere. Instead, after introducing the package basics, we focus on analysis of some real examples kindly provided by course participants, and thus reflecting current issues encountered by biometricians in Australasia.

The *asreml* function in R calls the core of the standalone ASReml package (Gilmour et al., 2009), which is also called by the REML facilities in GenStat (VSN International, 2012). In this workshop we focus on the functionality available in this R package in the context of the examples provided by workshop participants.

More detailed documentation of the *asreml* class and package can be found in the manual (Butler et al., 2009). This is file asreml-R.pdf located in directory *doc* in the *asreml* library directory. Note that the R command *.libPaths()* can be used to identify possible locations for this library. The manual can be accessed from within R using the command *asreml.man()* once the *asreml* library has been loaded via *library(asreml)*. Information on the version of *asreml* library present can be obtained using the command *asreml.About()*.

The *asreml* package requires a license to call the ASReml core functions. A trial license is available for workshop participants and has been distributed by email. Details of the license installed are included as a component of the *asreml* object (see Section 1.4).

These notes, the data files and R scripts are available from the conference website. Details will be circulated to participants directly.

1 The linear mixed model

In this chapter, we aim to give a gentle introduction to the class of linear mixed models covered by this workshop and to the ASReml package we will use to fit them. We start by introducing the generic form of the linear mixed model (Section 1.1). We then focus more specifically on the form used by ASReml and explain the rationale and implications of this choice (Section 1.2), with particular emphasis on the problem of model identifiability. Specification of different aspects of the model using the *asreml* function are then discussed, including how the model definition translates into algebraic form (Section 1.3). The *asreml* class of functions and its associated methods are then described a little more formally (Section 1.4). Finally, we provide some supplementary information regarding types and combinations of variance models within the random model or residual term (Section 1.5).

1.1 The linear mixed model

If $\boldsymbol{y}$ $(n \times 1)$ denotes the vector of observations, the linear mixed model can be written as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\tau} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e} \tag{1.1.1}$$

where $\boldsymbol{\tau}$ $(p \times 1)$ is the vector of fixed effects, $\boldsymbol{X}$ $(n \times p)$ is the design matrix that associates observations with the appropriate combination of fixed effects, $\boldsymbol{u}$ $(q \times 1)$ is the vector of random effects, $\boldsymbol{Z}$ $(n \times q)$ is the design matrix which associates observations with the appropriate combination of random effects, and $\boldsymbol{e}$ $(n \times 1)$ is the vector of residual errors.

The model (1.1.1) is called a linear mixed model or linear mixed-effects model. It is assumed

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{e} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{G}(\boldsymbol{\sigma}_g) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}(\boldsymbol{\sigma}_r) \end{bmatrix} \right) \tag{1.1.2}$$

where the matrices $\boldsymbol{G}$ and $\boldsymbol{R}$ are variance matrices for $\boldsymbol{u}$ and $\boldsymbol{e}$ that are functions of parameters $\boldsymbol{\sigma}_g$ and $\boldsymbol{\sigma}_r$. This requires that the random effects $\boldsymbol{u}$ and residual errors, $\boldsymbol{e}$, are uncorrelated. The variance matrix for $\boldsymbol{y}$ then takes the form

$$\text{var}\,(\boldsymbol{y}) \quad = \quad \boldsymbol{Z}\boldsymbol{G}(\boldsymbol{\sigma}_g)\boldsymbol{Z}' + \boldsymbol{R}(\boldsymbol{\sigma}_r) \tag{1.1.3}$$

and the estimates of $\boldsymbol{\sigma}_g$ and $\boldsymbol{\sigma}_r$ are referred to as *estimates on the component scale*. The variance models given by the matrices $\boldsymbol{G}$ and $\boldsymbol{R}$ are called G-structures and R-structures respectively.

We now consider an example to put these concepts into a practical context.

Example 1.1 An introductory example: malting quality of barley.

The data shown in Table 1.1 is taken from a study undertaken to examine the sources of variation on the assessment of barley malting quality. An important trait in determining the malt quality of barley is the diastatic power (DP). Samples of barley grain are put through a malting process using a micro-malter and DP is measured on these samples. The micro-malter holds 80 cannisters in a 16×5 array. Often there are more than 80 samples to be malted, so that sequential runs of the micro-malter must be undertaken. In this study 10 sequential malt runs were required. Four out of 80 cannisters in each run were randomly assigned to a standard barley variety. Each of these cannisters was filled with a subsample from a large batch of grain from the same variety. We examine these data to determine the relative importance of between malt run variation versus variation due to other sources. The 40 samples have a natural internal structure, namely cannisters within a malt run and malt runs. The "block" structure is therefore equivalent to a randomised block design with malt runs as blocks and cannisters as plots.

	Can 1	Can 2	Can 3	Can 4
Run 1	10.0	9.9	10.1	10.6
Run 2	9.1	10.3	10.0	9.0
Run 3	11.5	11.3	11.6	11.3
Run 4	10.0	9.6	10.6	10.8
Run 5	10.0	9.2	10.6	9.2
Run 6	10.0	10.9	10.9	10.1
Run 7	9.1	9.1	9.3	9.0
Run 8	9.0	8.3	9.9	10.0
Run 9	10.3	9.0	9.0	9.7
Run 10	9.1	9.1	8.9	9.0

Table 1.1: Diastatic power of control samples

This data is held in file *malt.dat* as three comma-separated columns with a header row giving the variable names (*Run* for the different runs, *Cannister* to identify the four different cannisters in each run and *dp* holding the DP values). The top 10 rows of the file take the form:

```
Run,Cannister,dp
1,1,10
1,2,9.9
1,3,10.1
1,4,10.6
2,1,9.1
2,2,10.3
2,3,10
2,4,9
3,1,11.5
```

Reading in the data

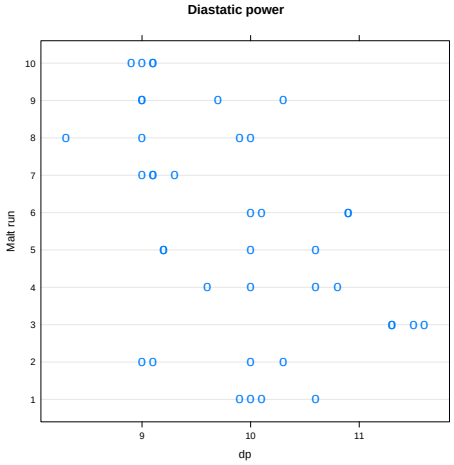

Figure 1.1: Dot plot of DP values for 4 cannisters in each of 10 runs.

We can set up the job and read this data set into R using the following commands

```
library(asreml)
malt <- asreml.read.table("malt.dat",sep=',',header=T)
```

The *asreml.read.table* command reads data from a text file and constructs a data frame. A header line containing variable names can be specified (*header*=T). Fields are initially read in as character data. If all the items in a field are numeric, the corresponding variable is converted to numeric. Otherwise, it is by default an (unordered) factor, although this can be over-ridden by the *as.is* argument, as per *read.table()*. Finally, fields whose name begins with a capital letter are converted to factors. Here, only *dp* is retained as a numeric variable, both the others are converted to factors since their names are capitalized.

We now have a data frame *malt*, with 2 columns as factors, *Run* and *Cannister* with 10 and 4 levels respectively and 1 column as a variate, *dp*. A useful graphical representation of this type of data (see Figure 1.1) can be produced using the *dotplot* command

```
dotplot(mrun ~ dp, data= malt, ylab='Malt run', main='Diastatic power', cex=2, pch="o")
```

The dotplot shows that the values from some runs are consistently lower than those from others, suggesting that runs contribute a substantial component of the total variation. We will quantify this through our analysis.

The statistical model

A simple statistical model that allows for both between and within malt run variation is

$$y_{ij} = \mu + u_i + e_{ij} \tag{1.1.4}$$

where y_{ij} is the observed DP, μ represents the mean DP across all malt runs, u_i represents the ith malt run effect, $i = 1, 2, \ldots, 10$, and the e_{ij} are residual errors, where we assume $e_{ij} \sim N(0, \sigma^2)$

for $i = 1, \ldots, 10$ and $j = 1, 2, \ldots, 4$. The aim of this analysis is to quantify variation across maltruns and so it is appropriate to assume the run effects are random with $u_i \sim N(0, \sigma_u^2), i = 1, \ldots, 10$. Both sets of effects (the malt run effects (u_i) and the residual errors (e_{ij})) are therefore independent and identically distributed (IID), each set having a common variance. In addition we assume the two sets are statistically independent. The parameters to be estimated are now μ (the mean DP over all malt runs), and the variance parameters σ_u^2 and σ_e^2.

If $\boldsymbol{y}$ is the vector of observations (ordered as cannisters within malt run), $n = 40$ is the sample size, $b = 10$ is the number of malt runs, and $t = 4$ is the number of cannisters per malt run, (1.1.4) can be written as

$$\boldsymbol{y} = \mathbf{1}_n \mu + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{e} \tag{1.1.5}$$

where $\mathbf{1}_n$ is a vector of n ones, $\boldsymbol{Z} = \boldsymbol{I}_b \otimes \mathbf{1}_t$ is the $n \times b$ design matrix for the vector of malt run effects, $\boldsymbol{u}$ is the vector of malt run effects, and we assume $\boldsymbol{u} \sim N(\mathbf{0}, \sigma_u^2 \boldsymbol{I}_b)$ and $\boldsymbol{e}$ is the vector of residual errors, and $\boldsymbol{e} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I}_n)$. If we set $\boldsymbol{X} = \mathbf{1}_n$ and $\boldsymbol{\tau} = \mu$ then this is in the form of Equation (1.1.1). Note that the symbol $\otimes$ is the direct (or kronecker) product operator: for any $(m \times p)$ matrix $A = [a_{ij}]$ for $i = 1 \ldots m$, $j = 1 \ldots p$, and any matrix $(n \times q)$ B, the direct product of A and B is an $(mn \times pq)$ matrix defined as

$$A \otimes B \quad = \quad \begin{bmatrix} a_{11}B & \ldots & a_{1p}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \ddots & a_{mp}B \end{bmatrix}.$$

We can derive the marginal distribution of $\boldsymbol{y}$ as

$$\boldsymbol{y} \sim N(\mathbf{1}_n \mu, \sigma_u^2 \boldsymbol{Z}\boldsymbol{Z}' + \sigma_e^2 \boldsymbol{I}_n) \tag{1.1.6}$$

The aim is to estimate σ_u^2 and σ_e^2 in order to gauge the relative size of between and within malt run variation. If we define the model directly in terms of the random effects, as in (1.1.5), then the parameters σ_u^2 and σ_e^2 occur explicitly as variances and must remain positive. If we define the model via its marginal form, as in (1.1.6), then the only requirement on the parameters σ_u^2 and σ_e^2 is that the overall variance matrix

$$\text{var}\,(\boldsymbol{y}) \quad = \quad \sigma_u^2 \boldsymbol{Z}\boldsymbol{Z}' + \sigma_e^2 \boldsymbol{I}_n \tag{1.1.7}$$

remains positive definite. For this specific example, this is true if both $\sigma_e^2 > 0$ and $t\sigma_u^2 + \sigma_e^2 > 0$. The latter condition allows σ_u^2 to be negative within a permitted range that is dependent on the value of σ_e^2.

Mapping this model back to the form of equation (1.1.3), we have $\boldsymbol{G}(\boldsymbol{\sigma}_g) = \sigma_u^2 \boldsymbol{I}_b$ and $\boldsymbol{R}(\boldsymbol{\sigma}_r) = \sigma_e^2 \boldsymbol{I}_n$ with parameter vectors $\boldsymbol{\sigma}_g = (\sigma_u^2)$, $\boldsymbol{\sigma}_r = (\sigma_e^2)$.

Fitting the model

We can fit this model using *asreml* and examine the estimated variance parameters via the commands

```
malt.asr <- asreml(fixed=dp~1, random=~idv(Run), rcov=~idv(units), data=malt)
summary(malt.asr)$varcomp
```

The *fixed* argument is used to specify both the vector of responses (*dp*) and the fixed model (here, only the intercept term μ, associated with $\mathbf{1}_n$, specified in *asreml* as *1*). The *random* argument is used to specify the random model (here the set of random run effects, $\boldsymbol{u}$, specified using the associated factor *Run*), and any variance models (here *idv()* specifies a set of IID effects with a common variance). The *rcov* argument is used to specify the variance model for the residual term, again a set of IID effects with common variance; the structure *units* is a reserved name for an internally-defined factor with a separate level for each unit of the data set. Finally, the *data* argument specifies the data frame holding the variables. The returned object *malt.asr* is of the *asreml* class (described in Section 1.4), and the *varcomp* element of the *summary* function here returns the estimated variance parameters.

In fact, we can specify this model more simply by using the default variance models, and we discuss this further in Sections 1.3.2 and 1.3.4.

asreml output

The output from fitting this model is shown below:

```
> malt.asr <- asreml(fixed=dp~1, random=~idv(Run), rcov=~idv(units), data=malt)

asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik          S2      DF      wall     cpu
    -82.5869      1.0000    39  04:55:45     0.0
    -58.5222      1.0000    39  04:55:45     0.0
    -32.4586      1.0000    39  04:55:45     0.0
    -12.3587      1.0000    39  04:55:45     0.0
     -5.9847      1.0000    39  04:55:45     0.0
     -4.7378      1.0000    39  04:55:45     0.0
     -4.6479      1.0000    39  04:55:45     0.0
     -4.6471      1.0000    39  04:55:45     0.0
     -4.6471      1.0000    39  04:55:45     0.0

Finished on: Tue Nov 20 04:55:45 2012

LogLikelihood Converged
> summary(malt.asr)$varcomp
                gamma component  std.error  z.ratio constraint
Run!Run.var 0.4706667 0.4706667 0.25323530 1.858614   Positive
R!variance  1.0000000 1.0000000         NA       NA      Fixed
R!units.var 0.2613333 0.2613333 0.06747598 3.872983   Positive
```

The *asreml* function returns summary output on the model fitting process, in terms of the residual log-likelihood function at each iteration, plus a statement of whether the model has converged. If the model has not converged, this may indicate problems and should be investi-

gated further before proceeding with model interpretation. In our example above, convergence has been successful. The estimated variance parameters (taken from the component column) are $\hat{\sigma}_u^2 = 0.471$ (labelled *Run!Run.var* indicating the variance of term Run) and $\hat{\sigma}_e^2 = 0.261$ (labelled *R!units.var* indicating the variance of the *units* factor in the R matrix). (We discuss the term labelled *R!variance* in Section 1.2). So it appears that variation due to run effects is almost twice the variation due to within-run effects.

□

Partitioning the fixed, random and residual models

Typically, the fixed and random models are each composed from several model terms. In this case it is natural to partition the fixed and random effects into sub-vectors corresponding to these terms. So then $\boldsymbol{\tau} = [\boldsymbol{\tau}'_1 \dots \boldsymbol{\tau}'_t]'$ and $\boldsymbol{u} = [\boldsymbol{u}'_1 \dots \boldsymbol{u}'_b]$ say, with $\boldsymbol{X}$ and $\boldsymbol{Z}$ partitioned conformably, with $\boldsymbol{X} = [\boldsymbol{X}_1\ \boldsymbol{X}_2\ \dots\ \boldsymbol{X}_t]$ and $\boldsymbol{Z} = [\boldsymbol{Z}_1\ \boldsymbol{Z}_2\ \dots\ \boldsymbol{Z}_b]$.

In parallel, we impose a direct sum structure on the matrix $\boldsymbol{G}$, written

$$\boldsymbol{G} = \oplus_{i=1}^{b'} \boldsymbol{G}_i = \begin{bmatrix} \boldsymbol{G}_1 & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{G}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{G}_{b'-1} & 0 \\ 0 & 0 & \dots & 0 & \boldsymbol{G}_{b'} \end{bmatrix}$$

where $\oplus$ is the direct sum operator.

The default assumption is that each random model term generates one component of this direct sum (then $b' = b$ and $\text{var}(\boldsymbol{u}_i) = \boldsymbol{G}_i$ for $i = 1 \dots b$). This means that the random effects from any two distinct model terms are uncorrelated. However, in some models correlation is required across model terms (eg. in random coefficient regression, the random intercepts and slopes for subjects are correlated). To accommodate these cases, one component of $\boldsymbol{G}$ may apply across several (adjacent) random model terms (then $b' < b$).

We allow the matrix R to also have a direct product structure, with

$$\boldsymbol{R} = \oplus_{j=1}^{s} \boldsymbol{R}_j = \begin{bmatrix} \boldsymbol{R}_1 & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{R}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{R}_{s-1} & 0 \\ 0 & 0 & \dots & 0 & \boldsymbol{R}_s \end{bmatrix}$$

for $s \geq 1$ where the units corresponding to individual components of this direct sum structure are called sections. Note that it may be necessary to re-order (re-number) the data units in order to achieve this structure. As for $\boldsymbol{G}$, each component of the direct product structure for $\boldsymbol{R}$ can be defined independently, ie. using a separate variance model.

In many cases, the full set of residual errors ($\boldsymbol{e}$) can be expected to share a single variance model, and then there is only one section ($s = 1$).

However sometimes it is natural to partition the data into sections according to some criterion (factor), such that the errors corresponding to each section would be expected to require different variance structures. For example, in analysis of a set of multi-environment trials, it is natural to expect that each trial will require a separate (possibly spatial) error structure. This facility can also be used to allow for different residual variances across distinct sets of units.

Thus in general we order the data units so that we can write $\boldsymbol{e} = [\boldsymbol{e}_1', \boldsymbol{e}_2', \ldots, \boldsymbol{e}_s']'$ so that $\boldsymbol{e}_j$ represents the vector of errors of the jth *section* of the data, and sections can be assumed independent with $\text{var}(\boldsymbol{e}_j) = \boldsymbol{R}_j$.

1.2 Introducing an overall scale parameter (θ)

Model (1.1.3) is the simplest parameterization of $\text{var}(\boldsymbol{y})$. A more general form that fits an equivalent model uses

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{e} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \theta \begin{bmatrix} \boldsymbol{G}(\boldsymbol{\gamma}_g) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}(\boldsymbol{\gamma}_r) \end{bmatrix} \right) . \tag{1.2.8}$$

The parameter θ is an overall scaling parameter that affects both the estimated value and the interpretation of some (but not necessarily all) of the variance parameters. To reinforce this point, we label the variance parameters here as θ plus $\boldsymbol{\gamma}_g$ and $\boldsymbol{\gamma}_r$. The estimates of $\boldsymbol{\gamma}_g$ and $\boldsymbol{\gamma}_r$ are referred to as *estimates on the ratio scale*. Now

$$\text{var}(\boldsymbol{y}) \quad = \quad \theta \left(\boldsymbol{Z}\boldsymbol{G}(\boldsymbol{\gamma}_g)\boldsymbol{Z}' + \boldsymbol{R}(\boldsymbol{\gamma}_r) \right) . \tag{1.2.9}$$

This is the model definition currently used by the *asreml* function.

There are both advantages and disadvantages associated with this parameterization. The main advantage is that estimation of θ puts variance parameters onto the ratio scale, where it can be easier to specify initial values (eg. for variance ratios). The main disadvantage is in potential confusion for the user, as not all of the variance parameters are re-scaled, and output that contains estimates on both the component and ratio (gamma) scale can be difficult to follow. We feel that the disadvantages now outweigh the advantages, and future releases of ASReml will by default present estimates on the component scale (although estimation within the underlying algorithm may still proceed on the components scale for reasons of computational efficiency).

In fact, we can be more specific about the ways in which different variance parameters are affected by the presence of θ by partitioning the variance parameters associated with $\boldsymbol{G}$ and $\boldsymbol{R}$ into those that define variance/covariances (and hence are scale-dependent), and those that define correlations (and so are scale-independent), as shown in Table 1.2.

The scale-independent parameters (eg. the correlation parameter of an AR(1) process) are unaffected by the presence of θ, so that

$$\boldsymbol{\sigma}_{gc} = \boldsymbol{\gamma}_{gc}; \qquad \boldsymbol{\sigma}_{rc} = \boldsymbol{\gamma}_{rc}.$$

Table 1.2: Partitioning parameter sets into those that define variance/covariances (scale-dependent) and those that define correlations (scale-independent, ie. parameters of correlation matrices).

Parameterization	Model	Set	Variance/covariance	Correlation
Without θ	$\boldsymbol{G}$	$\boldsymbol{\sigma}_g$	$\boldsymbol{\sigma}_{gv}$	$\boldsymbol{\sigma}_{gc}$
Without θ	$\boldsymbol{R}$	$\boldsymbol{\sigma}_r$	$\boldsymbol{\sigma}_{rv}$	$\boldsymbol{\sigma}_{rc}$
With θ	$\boldsymbol{G}$	$\boldsymbol{\gamma}_g$	$\boldsymbol{\gamma}_{gv}$	$\boldsymbol{\gamma}_{gc}$
With θ	$\boldsymbol{R}$	$\boldsymbol{\gamma}_r$	$\boldsymbol{\gamma}_{rv}$	$\boldsymbol{\gamma}_{rc}$

The interpretation of the scale-dependent parameters does change when θ is included in the model. The effect of θ depends on the parameter type. In most cases, the scale-dependent parameter is a variance or covariance parameter and the parameter becomes a ratio with θ, so that

$$\boldsymbol{\sigma}_{gv} = \theta\boldsymbol{\gamma}_{gv}; \qquad \boldsymbol{\sigma}_{rv} = \theta\boldsymbol{\gamma}_{rv}.$$

For the few exceptions (loadings for factor analytic (*fa*) models, diagonal components of ante-dependence (*ante*) models), the parameter becomes either a ratio with respect to $\sqrt{\theta}$ (*fa* loadings with $\boldsymbol{\sigma}_{gv} = \sqrt{\theta}\boldsymbol{\gamma}_{gv}$) or a multiple of θ (*ante* diagonal matrix with $\theta\boldsymbol{\sigma}_{gv} = \boldsymbol{\gamma}_{gv}$).

Example 1.2 Malt run example revisited with θ

We can re-write the variance structure for the one-way model as

$$\text{var}\,(\boldsymbol{y}) \quad = \quad \theta(\gamma_u \boldsymbol{Z}\boldsymbol{Z}' + \boldsymbol{I}_n) \tag{1.2.10}$$

using $\boldsymbol{u} \sim N(\boldsymbol{0}, \theta\gamma_u \boldsymbol{I}_b)$ where $\boldsymbol{\gamma}_u = \sigma^2_u/\sigma^2_e$ and $\boldsymbol{e} \sim N(\boldsymbol{0}, \theta\boldsymbol{I}_n)$ where $\theta = \sigma^2_e$. So this form also has two parameters and can be mapped directly onto the original parameterization. The two parameters are the ratio of the random effects variance to the residual variance (γ_u) plus the residual variance itself (now represented as θ). In terms of equation (1.2.8), we have $\boldsymbol{G}(\boldsymbol{\gamma}_g) = \gamma_u \boldsymbol{I}_b$ and $\boldsymbol{R}(\boldsymbol{\gamma}_r) = \boldsymbol{I}_n$ with $\boldsymbol{\gamma}_g = (\gamma_u)$ and $\boldsymbol{\gamma}_r$ is a null vector (containing no parameters). All of the parameters here are scale-dependent.

We can fit this model using *asreml* by replacing *idv(units)* in our previous call (Example 1.1) by *id(units)*. The commands to fit the model examine the results become

```
malt.asr2 <- asreml(fixed=dp~1, random=~idv(Run), rcov=~id(units), data=malt)
summary(malt.asr2)$varcomp
```

The output from the summary function presents the following set of estimated variance parameters:

```
> summary(malt.asr2)$varcomp
              gamma component  std.error  z.ratio constraint
Run!Run.var 1.80102 0.4706667 0.25323530 1.858614   Positive
R!variance  1.00000 0.2613333 0.06747598 3.872983   Positive
```

There are two variance parameters labelled as *Run!Run.var*, ie. the variance of the Run term,

and *R!variance*, which is θ. The gamma column gives the estimates on the ratio scale, with $\hat{\gamma}_u = 1.80$. A dummy value of 1.0 is always given for θ in this column. The component column rescales the estimates onto the component scale, giving $\hat{\theta} = 0.261$ and $\hat{\sigma}^2_u = \hat{\theta}\hat{\gamma}_u = 0.471$, as previously.

The issue of identifiability

Although we did not appear to explicitly request the presence of θ in the model here, nevertheless it was present because the current default model definition for *asreml* includes θ and it had not been suppressed. In general, θ is suppressed (in fact θ is fixed at value 1) if the *asreml* function determines that estimation of this parameter would lead to aliasing (variance parameters not identifiable). This suppression happened in Example 1.1 because an identity structure with common variance was requested for the residual and the additional estimation of θ would have led to unidentifiability.

To illustrate the issue of identifiability in more detail, we write down an extended version of this model (which might be considered the most general parameterization) as

$$\text{var}\,(\boldsymbol{y}) \quad = \quad \theta(\gamma_u \boldsymbol{Z}\boldsymbol{Z}' + \gamma_e \boldsymbol{I}_n) \tag{1.2.11}$$

by re-introducing a scalar, γ_e, into the residual variance term. The variance matrix for the random effects ($\boldsymbol{u}$) is unchanged (equal to $\theta\gamma_u \boldsymbol{I}_b$). The variance of the residual errors becomes $\text{var}\,(\boldsymbol{e}) = \theta\gamma_e \boldsymbol{I}_n$. This model has three variance parameters: θ, $\boldsymbol{\gamma}_g = (\gamma_u)$ and $\boldsymbol{\gamma}_r = (\gamma_e)$. This model is unidentifiable because there are an infinite numbers of ways in which to achieve the same result: for example, if we double θ, we merely have to halve γ_u and γ_e to get back to the same variance model. The REML algorithm is only reliable when all of the variance parameters are uniquely identifiable. Here, we can achieve this by fixing either $\gamma_e \equiv 1$ or $\theta \equiv 1$. In general, problem of identifiability can occur whenever several scale-dependent parameters are multiplied together. Where it can, the *asreml* function will resolve these conflicts by imposing constraints to make the variance model identifiable. However, this cannot always be done automatically, and it is important that users of complex models understand the rules discussed in Section 1.5. The default constraints used by *asreml* are context-dependent and are discussed in Sections 1.3.2 and 1.3.4.

□

1.3 Model specification using ASReml

As seen in the previous Section, a call to the *asreml* function takes the generic form

```
fit.asr <- asreml(fixed=y~model formula, random=~model formula, rcov=~model formula,
  data=dataframe)
```

The model formulae are specified using an extended version of the syntax defined in the seminal paper by Wilkinson & Rogers (1973), which uses the crossing operator * and nesting operator /. For factors A and B, these operators are defined as $A*B = A + B + A:B$ and $A/B = A +$

Table 1.3: List of common constructor functions used in asreml (x is a variate, A is a factor).

Function name	Model Fixed/random	Description
dev(x)	Either	Factor version of variate x
lin(A)	Either	Variate version of factor A (using ordinal levels)
pol(x,t)	Either	Orthogonal polynomials of order 0:t generated from x
pol(x,-t)	Either	Orthogonal polynomials of order 1:t generated from x
spl(x)	Random	Non-linear component of a cubic spline for variate x
at(A,l)	Either	Condition on lth level of factor A, where l may be a list of levels. Used to include model terms for a subset of factor levels.
and()	Either	Allows construction of non-standard design matrices

A:B, where *A:B* consists of all combinations of levels from the factors A and B. The model is processed to give a set of *model terms* linked by the + operator. Model terms are constructed from individual variables, which in general may be factors or variates.

1.3.1 Fixed and random model terms

Fixed model terms are used to construct the design matrix $\boldsymbol{X}$ and associated effects $\boldsymbol{\tau}$, and random model terms are used to construct the design matrix $\boldsymbol{Z}$ and associated effects $\boldsymbol{u}$ given in Equation (1.1.1).

Functions can be applied to the variables comprising the fixed or random model terms to modify the form of the term. We call these *constructor functions*, as they are used to modify the construction of the design matrices $\boldsymbol{X}$ and $\boldsymbol{Z}$. The most common constructor functions are listed in Table 1.3; full details of these and additional functions (including *grp()* and *mbf()*) can be found in the ASReml manual.

1.3.2 Adding variance models into random terms

The random model terms define the design matrix and associated set of random effects, but additional information is required to define the G-structure for each term. In general, there are two different approaches to this problem, which we will call the functional and structural approaches. The structural approach applies variance models to individual terms after the random model has been defined; this is the approach taken in the GenStat and ASReml-SA (standalone package) interfaces to the ASReml core. This gives a step-by-step approach to building up the full model that is straightforward but not concise. In contrast, the functional approach uses functions to directly apply variance models to individual variables in the random model terms to produce a consolidated model term that simultaneously defines both the design matrix ($\boldsymbol{Z}_i$) and variance model ($\boldsymbol{G}_i$). This process is shown in Table 1.4 and some common variance functions are defined in Table 1.5. The full range of variance model functions and their detailed definition can be found in the ASReml manual.

Table 1.4: Building a consolidated model term

Model term	Variables in term	Variance model function	Covariance component	Consolidated term
A:B	*A*	*id()*	*id(A)*	*id(A):ar1(B)*
	B	*ar1()*	*ar1(B)*	

Table 1.5: List of common variance model functions, their type (correlation or variance) and a brief description.

Function name	Type	Description
id()	Correlation	IID with variance 1
idv()	Variance	IID with common variance
idh()	Variance	independent with separate variances
ar1()	Correlation	auto-regressive structure of order 1
cor()	Correlation	unstructured correlation matrix
exp()	Correlation	exponential power model (based on distances)
diag()	Variance	independent with separate variances (same as *idh()*)
us()	Variance	general unstructured covariance matrix

In Table 1.5, the correlation models take value 1 on the diagonal but variance models can take any positive value on the diagonal. Names of correlation models can be appended with *v* (eg. *idv()*) to add a common (homogeneous) variance or with *h* (eg. *idh()*) to add a separate (heterogeneous) variance for each level of the factor. For these models, the parameters associated with the correlation model are scale-independent, and those associated with the common or heterogeneous variances are scale-dependent.

If the consolidated model term definition is incomplete, ie. if some (or all) of the variables do not have a variance model specified, the default variance model, *id()*, will be applied to these variables.

Once all variables have a variance model function applied, we can assess whether the term is identifiable. If the consolidated term generates a correlation matrix, then it is usually the case that one wishes to fit a model with this specified correlation structure but to also allow the effects have a common variance. In certain cases, the *asreml* function will detect this and add a common variance. In general, it is necessary for the user to complete the specification. So for example *id(A):ar1(B)* should become either *idv(A):ar1(B)* or *id(A):ar1v(B)*; it is arbitrary which variable the common variance is attached to. If more than one function generates a variance model (either homogeneous or heterogeneous), then the parameters will not all be identifiable and the user must impose a suitable set of constraints (see Section 1.5).

A consolidated model term thus consists of a set of variables, each with a variance model function applied. This generates a *separable* variance structure for the term.

1.3.3 Separability

The concept of separability has been used extensively in multivariate analysis of variance and was described by Martin (1979) in the context of lattice processes. Martin (1979) showed that the correlation matrix of a linear-by-linear process observed on a $r \times c$ rectangular lattice can be written as the kronecker (or direct) product of two correlation matrices which relate to the rows and columns of the lattice. Both formulations are used frequently in the context of linear mixed models.

In the general case, consider a term *A:B*, where factors A and B have n_A and n_B levels respectively. Suppose the consolidated model term is of generic form *f(A):g(B)*, where *f(A)* generates a variance matrix of form V_A and *g(B)* generates a variance matrix of form V_B. The effects associated with this term $\boldsymbol{u}_{A:B} = (u_{1,1}\, u_{1,2} \ldots u_{1,n_B}\, u_{n_A,n_B-1}\, u_{n_A n_B})'$ then have variance matrix

$$\text{var}\,(\boldsymbol{u}_{A:B}) = V_A \otimes V_B$$

so that the covariance between individual random effects is

$$\text{cov}\,(u_{ij},\ u_{kl}) = [V_A]_{i,k} \times [V_B]_{j,l}$$

ie. the covariance is equal to that generated between levels i and k of factor A by model V_A (ignoring factor B), multiplied by the covariance generated between levels j and l of factor B by model V_B (ignoring factor A).

The assumption of separability greatly reduces the computational load and contributes to the efficiency of the ASReml core. However, separability also allows a flexible framework for modelling variance structures in the linear mixed model that is genuinely appropriate in many situations. Let us consider two common examples:

1. Multivariate analysis. In multivariate analysis of several traits, it is common to assume independence between subjects, but to allow correlation across traits at all levels of the structure. At the residual level, this can be generated by a separable structure defined using the consolidated model term *id(subject):us(trait)*. Analogous definitions can be used at other levels of the structure. In addition, correlation between subjects, eg. as a result of genetic relationships, can be applied using a suitable variance model function for the *subject* term.

2. Spatial analysis of a field trial. Field trials are often laid out on a grid pattern (rows $\times$ columns), with various management operations being aligned with either rows or columns of the grid. It is natural to assume this may induce correlations across rows within columns and vice versa, leading to a separable correlation structure defined by *ar1(row):ar1(col)*. Note that this correlation structure can be converted into a variance structure (common variances) by changing one of the variance models *ar1* into *ar1v*; this can arbitrarily be applied to either component of this term.

1.3.4 The model for the residual error term

Definition of the residual error term is also achieved via a consolidated model term. However, this term has some special properties that deserve separate consideration.

The size of the residual term (number of effects) must be equal to the number of data units included in the analysis, and each combination of levels of the factors comprising the term must uniquely identify one unit of the data. In addition, the data must be ordered to match the R-structure specified. These conditions will always be satisfied for a single section with IID errors, but a mismatch in both size and ordering is possible when either multiple sections are present or when non-identity variance models are used.

We will consider these issues in the context of a set of field trials. We first consider analysis of a single section: one trial with 4 replicates of 24 varieties arranged on a grid with 4 rows and 24 columns (rows are replicates). The data frame takes the form:

```
Trial  Row  Column Variety yield
1      1    1      C        5.43
1      1    2      M        6.01
1      1    3      J        6.31
.
.
.
1      4    23     R        4.22
1      4    24     B        4.89
```

where *Trial*, *Row*, *Column* and *Variety* are factors and *yield* is a variate. To get a separable auto-regressive spatial model (order 1), we can specify the residual term as

```
rcov=~ar1(Row):ar1(Column)
```

The ordering of the two terms is important: this specification means that the errors are ordered as columns within rows. As there is no design matrix to mediate between the data and the vector of residual errors, it is necessary for the data to also be ordered as columns within rows (as is the case here). In the current version of *asreml*, this is checked and a message is printed if the ordering of the data appears incompatible with the model specified.

An error will also be given if the number of data units is not equal to the number of effects generated by the direct product structure, ie. the product of the number of levels in the factors used to construct the term. If this is the case, then the term specified cannot be used as the residual term.

For a single section, if the consolidated model term generates a correlation model then the overall scaling parameter θ will be estimated, so that parameters are estimated on the ratio scale. Otherwise, θ will be fixed at one and parameters estimated on the component scale.

We next consider the case of multiple sections via joint analysis of a series of 12 field trials on the same 24 varieties held in a single data frame. All the trials use a row by column grid layout but with different shapes on some sites (trials 1-8 are 4×24 but trials 9-12 are 8×12). The data is ordered by trial, then by columns within rows for each trial. If we wish to fit a separable auto-regressive model (order 1) within each trial, we can specify

```
rcov=~at(Trial):ar1(Row):ar1(Column)
```

If we wish to fit an autoregressive structure in the column direction only for trials 1-8 (with independence across rows), we can specify

```
rcov=~at(Trial,c(1:8)):id(Row):ar1(Column)+at(Trial,c(9:12)):ar1(Row):ar1(Column)
```

In the context of an *rcov* definition, the *at()* function performs several different tasks. The underlying *Trial:Row:Column* term tells *asreml* that we are working with a direct product of the *Trial*, *Row* and *Column* factors. The use of *at(Trial)* modifies this to a direct sum structure, with the different components of the direct sum (sections) applying to units with different levels of the factor *Trial*. The use of *at(Trial)* in this context also means that the *Row:Column* products are pruned within trials (so that only the levels used within each trial apply to that trial) and a separate residual variance is applied to each trial. Again, where the within-section models are defined as correlation matrices (as above), a common variance will automatically be added within each section.

For each section (here trials), it is necessary that the size of the direct product structure generated (after pruning) matches the number of units in the section, and that the ordering of the units within each section matches that expected for the direct product structure (here, ordered as columns within rows for each trial).

This context-dependent behaviour of the *at()* function is undesirable as it is potentially confusing and this will be resolved in the next release of *asreml*, where it will revert to a pure constructor function. A new special function *dsum()* will become available to explicitly define the direct product structure of R.

When multiple sections are present, the overall scaling parameter θ is always fixed at value 1, so that estimates are made on the component scale. For sections defined in terms of correlation structures, a common variance will be added separately to each of those sections.

Example 1.3 Model specification for the malt run example

We are now in a position to understand the various different ways of specifying the model for the malt run example. Some of these are summarised in Table 1.6, numbered so we can identify them easily. The first model (Model 0) makes full use of the defaults, with specification as

```
malt.asr0 <- asreml(fixed=dp~1, random=~Run, data=malt)
```

Table 1.6: Selection of different (but equivalent) specifications of the model for the malt run data

Model	*random*=∼	*rcov*=∼	Status of θ
0	Run	-	Estimated
1	idv(Run)	idv(units)	Fixed at 1
2	idv(Run)	id(units)	Estimated
3	Run	Run:Cannister	Estimated
4	idv(Run)	Run:Cannister	Estimated
5	idv(Run)	id(Run):id(Cannister)	Estimated
6	idv(Run)	idv(Run):id(Cannister)	Fixed at 1
7	idv(Run)	id(Run):idv(Cannister)	Fixed at 1

The program interprets *random= Run* first as *id(Run)*, then converts this to *idv(Run)* as no scalar parameter is present for this random term. The residual term is absent, and so is effectively defined internally as *id(units)*, and θ is estimated to provide a scaling for this term.

Models 1 and 2 were considered earlier. Models 3-7 acknowledge the direct product structure of the residual term as *Run:Cannister*. As this term requires only a common variance, this can be added by default (Models 3-5), invoking the estimation of θ, or explicitly included via the *idv* function applied to just one of the factors (Models 6 and 7). In these latter cases, θ is fixed at 1. Beware that the specification *rcov = idv(Run):idv(Cannister)* would produce an over-parameterised residual term and an error would occur.

□

1.3.5 Imposing structure across random model terms

The special function *str()* can be used to make a component of the direct-sum G-structure apply across several model terms. This is most commonly required for random coefficient regression models, to enable correlation between random subject intercepts and slopes. In the simplest case, with a set of 5 subjects (factor *Subject*) with explanatory variate *x*, this is specified as:

```
random = ~ str(form=~ Subject/x, vmodel=~ us(2):id(5))
```

The algorithm places the model terms specified using the argument form together in the processed random model (here *Subject* followed by *Subject.x*). The variance structure defined using the *vmodel* argument begins at the start of the first term specified and is expected to exactly span the whole set of terms given in argument form. The overall size of the variance model is checked against the total number of levels of these terms, but the user must verify that the ordering is appropriate to the variance model required. In this context, the size of the variance structures can be given as an integer argument to the variance functions, as a suitable factor is usually not available.

In our example, this random model generates a combined set of random effects from the individual subject intercepts, $\boldsymbol{u}_I = (u_{I1} \dots u_{I5})'$ and subject slopes, $\boldsymbol{u}_S = (u_{S1} \dots u_{S5})'$, as

$\boldsymbol{u}_{IS} = (\boldsymbol{u}_I' \, \boldsymbol{u}_S')'$. The consolidated term then has variance structure of the form

$$\text{var}\,(\boldsymbol{u}_{IS}) = \text{var}\left(\begin{bmatrix}\boldsymbol{u}_I \\ \boldsymbol{u}_S\end{bmatrix}\right) = \begin{pmatrix}\sigma_{II} & \sigma_{IS} \\ \sigma_{IS} & \sigma_{SS}\end{pmatrix} \otimes \boldsymbol{I}_5 = \begin{pmatrix}\sigma_{II}\boldsymbol{I}_5 & \sigma_{IS}\boldsymbol{I}_5 \\ \sigma_{IS}\boldsymbol{I}_5 & \sigma_{SS}\boldsymbol{I}_5\end{pmatrix}$$

Here, the set of subject intercepts has a common variance (σ_{II}), and the set of subject slopes has a (different) common variance (σ_{SS}). Intercepts and/or slopes from two different subjects are independent, but the intercept and slope from any given subject have covariance σ_{IS} (or correlation $\sigma_{IS}/\sqrt{\sigma_{II}\sigma_{SS}}$).

1.4 The ASReml class

In the previous sections, we have given a brief illustration of how to use *asreml* to specify and fit models. Running the *asreml* function produces an object of the *asreml* class. The structure of this object can be examined using the *names* function, as below, and further details on these components are given in Chapter 7 of Butler et al. (2009).

```
> names(malt.asr)
 [1] "monitor"           "loglik"            "gammas"            "gammas.type"       "gammas.con"
 [6] "stratumVariances"  "score"             "coefficients"      "vcoeff"            "predictions"
[11] "residuals"         "linear.predictors" "hat"               "aom"               "sigma2"
[16] "deviance"          "nedf"              "ai"                "nwv"               "noy"
[21] "nolev"             "noeff"             "nsing"             "yssqu"             "Cfixed"
[26] "Csparse"           "aovTbl"            "converge"          "last.message"      "license"
[31] "ifault"            "call"              "distribution"      "link"              "family"
[36] "fitted.values"     "control"           "G.param"           "R.param"           "factor.names"
[41] "fixed.formula"     "random.formula"    "sparse.formula"
```

Individual elements of asreml objects can be accessed using eg. *malt.asr$loglik*. Objects of the *asreml* class have methods defined for various generic functions to enable common tasks. Some of the most common are outlined in Table 1.7 and some of those will be briefly illustrated here. Others may be used in later chapters. Further details on all aspects are available in Butler et al. (2009). Details of the ASReml license, including the expiry date, can be printed out for the asreml class objects (eg. *malt.asr* as generated above) using the command

```
cat(malt.asr$license)
```

The *summary()* function processes an *asreml* object to produce a set of summary statistics for the model fit, including the set of estimated variance parameters as shown in Example 1.1. If argument *all=True* then the full set of coefficients (eBLUEs and eBLUPs) are also included with their estimated SEs. Argument *nice=T* gives an improved output format for parameters for the more complex variance models.

The *plot()* function can be used to produce a set of residual plots, as shown for the malt run example in Figure 1.4. By default, these plots are based on simple residuals ($\tilde{e}$) and allow a visual assessment of normality, variance heterogeneity and serial correlation. More complex plots are also possible, eg. plotting fitted values against residuals with points coloured (or separated into

Table 1.7: Some commonly-used methods for objects of class *asreml*.

Method	Description
summary()	Gives summary of model fit and estimated variance parameters
plot()	Produces a set of residual plots
coef()	The full set of fixed and random effects.
	coef(object)$fixed gives the fixed effects only, conversely *coef(object)$random*.
fitted()	Fitted values from the fitted model.
residuals()	Residual from the fitted model, corresponding to rcov term.
	The *type* argument can be used to determine how the residuals are calculated.
wald()	Table of incremental Wald or F tests.
	Arguments are available to determine the form of the tests (sequential vs marginal) and the method for calculating the denominator degrees of freedom.
predict()	Table of predictions for factor combinations specified.
	Many arguments to manipulate the exact form of the predictions.

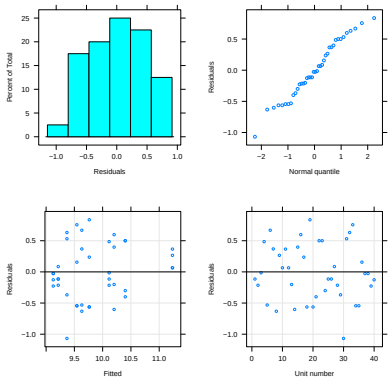

Figure 1.2: Set of residual plots for malt run example.

different panels) by treatment groups (or other factors).

The *coef()* function is used to obtain the estimated fixed effects (eBLUEs) and predicted random effects (eBLUPs). Argument *list=T* can be used to partition the sets of effects by model terms.

The *predict()* function is a very powerful method for producing predictions for specified combinations of variables from the fitted model. Predictions are formed by taking averages over fitted values from the predicted model, as described by Welham et al. (2004), and the predict function contains many arguments to control this averaging process. In the simplest case, it can be used to get a set of predictions with SEs for any factor in the model, averaged over levels of all other factors. The predict function produces a composite object, from which predictions and their SE can be extracted. In our malt run example, the commands

```
malt.pv <- predict(malt.asr, classify=Run)
malt.pv$predictions
```

produce the predictions (and associated information) in object malt.pv. The predictions component of this object contains the predicted values with their SE and an indication of whether each prediction is estimable (ie. invariant to parameterization of the fixed effects) as shown below, plus the average SED (calculated on the variance scale).

```
> malt.pv$predictions
$pvals

Notes:
- The predictions are obtained by averaging across the hypertable
  calculated from model terms constructed solely from factors in
  the averaging and classify sets.
- Use "average" to move ignored factors into the averaging set.


   Run predicted.value standard.error est.status
1    1       10.114652      0.2411766  Estimable
2    2        9.631692      0.2411766  Estimable
3    3       11.234241      0.2411766  Estimable
4    4       10.202463      0.2411766  Estimable
5    5        9.763408      0.2411766  Estimable
6    6       10.400037      0.2411766  Estimable
7    7        9.214590      0.2411766  Estimable
8    8        9.368259      0.2411766  Estimable
9    9        9.543881      0.2411766  Estimable
10  10        9.126779      0.2411766  Estimable

$avsed
  overall
0.3387324
```

Finally, the *wald()* function produces a table of Wald or approximate F statistics for fixed terms in the model. This is discussed further in Chapter 2.

1.5 More on variance models

In this supplementary section, we provide some more detailed background on different types of variance models. There are three generic types of variance model allowed in *asreml* that can be used for R and G-structures , namely, *correlation models*, *homogeneous variance models* and *heterogeneous variance models*. In the following sub-sections, we give a formal definition of each type of model, and then discuss how they can be combined to give valid variance models with an identifiable scale.

1.5.1 Correlation models:

In correlation models all diagonal elements are identically equal to 1. If the $n \times n$ matrix $\boldsymbol{C} = [c_{ij}]$, $i, j = 1, \ldots, n$, denotes the correlation matrix for a particular correlation model, then

$$\boldsymbol{C} = [c_{ij}] \quad : \quad \begin{cases} c_{ii} = 1, & \forall i \\ c_{ij} = c_{ji} & |c_{ij}| < 1,\ i \neq j. \end{cases}$$

The simplest correlation model is the identity model (*id()*) for which the off-diagonal elements are identically equal to zero, that is, $c_{ij} = 0$ for $i \neq j$.

Correlation models often arise in longitudinal data analysis (eg. *ar1()*), and in geostatistics or spatial statistics (eg. *exp()*, *mtrn()*). There are also more general correlation models such as the banded model which is a simplification of the completely general correlation model with $p(p-1)/2$ parameters.

1.5.2 Homogeneous variance models

In homogeneous variance models the diagonal elements all have the same positive value, σ^2 say. If $n \times n$ matrix $\boldsymbol{V} = [v_{ij}]$, $i, j = 1, \ldots, n$ is an homogeneous variance matrix, then

$$\boldsymbol{V} = [v_{ij}] \quad : \quad \begin{cases} v_{ii} = \sigma^2, & \forall i \\ v_{ij} = v_{ji}, & i \neq j. \end{cases}$$

Note that if $\boldsymbol{V}$ is the homogeneous variance model matrix corresponding to the correlation model matrix $\boldsymbol{C}$ then

$$\boldsymbol{V} \quad = \quad \sigma^2 \boldsymbol{C}$$

with $v_{ij} = \sigma_i^2 c_{ij}$. This model has just one more parameter than $\boldsymbol{C}$. For example, the homogeneous variance model corresponding to the identity correlation structure is the simple variance components model (*idv()*), for which $v_{ii} = \sigma^2$ for $i = 1 \ldots n$, with $v_{ij} = 0$ for $i \neq j$, ie. off diagonal elements equal to zero.

1.5.3 Heterogeneous variance models:

The third variance model is the *heterogeneous* variance model for which the diagonal elements are positive but differ. If $n \times n$ matrix $\boldsymbol{V} = [v_{ij}]$ for $i, j = 1, \ldots, n$, is a heterogeneous variance matrix, then

$$\boldsymbol{V} = [v_{ij}] \quad : \quad \begin{cases} v_{ii} = \sigma_i^2, & i = 1, \ldots, n \\ v_{ij} = v_{ji}, & i \neq j. \end{cases}$$

If $\boldsymbol{V}$ is the heterogeneous variance model matrix corresponding to the correlation model matrix $\boldsymbol{C}$, then

$$\boldsymbol{V} \quad = \quad \boldsymbol{DCD}$$

where $\boldsymbol{D}$ is a diagonal matrix with n rows, ie. $\boldsymbol{D} = \text{diag}\,(\sigma_i)$ and $v_{ij} = \sigma_i \sigma_j c_{ij}$. This model has an additional n parameters compared to the base correlation model. For example, the heterogeneous variance model corresponding to the identity correlation model (*idh()* or *diag()*) specifies the diagonal variance model for which $v_{ii} = \sigma_i^2$ for all $i1 \ldots n$, with zero off diagonal elements.

Other examples include the factor analytic (*fa*) or ante-dependence *ante* variance models and the most general is the unstructured (*us*) with $p(p+1)/2$ parameters.

1.5.4 Combining variance models

There are some general principles which can be useful in avoiding over-parameterisation of variance models and in the following we present some of these by way of example.

When either $\boldsymbol{R}$ or $\boldsymbol{G}$ is formed from the kronecker product of several sub-matrices some general rules must be obeyed to avoid over-parameterisation. In the following we consider models with two components for $\boldsymbol{G}$ and $\boldsymbol{R}$ and use $\boldsymbol{C}_i$ and $\boldsymbol{V}_i, i = 1, 2$ to denote arbitrary correlation and variance matrices, respectively.

1. If $\boldsymbol{R} = \boldsymbol{C}_1 \otimes \boldsymbol{C}_2$ then this is a valid correlation model and a scale parameter must be added. This can be done either by estimating θ or by fixing $\theta = 1$ and adding a common variance to one of the correlation models. The default action is described in Section 1.3.4.

2. If $\boldsymbol{G} = \boldsymbol{C}_1 \otimes \boldsymbol{C}_2$ then a scale parameter should be added to one of the correlation models. The default action is described in Section 1.3.2.

3. If $\boldsymbol{R} = \boldsymbol{C}_1 \otimes \boldsymbol{V}_2$ or $\boldsymbol{R} = \boldsymbol{V}_1 \otimes \boldsymbol{C}_2$ this defines a valid (identifiable) variance matrix. In this case θ cannot be estimated and will automatically be fixed at 1.

4. If $\boldsymbol{G} = \boldsymbol{C}_1 \otimes \boldsymbol{V}_2$ or $\boldsymbol{G} = \boldsymbol{V}_1 \otimes \boldsymbol{C}_2$ then $\boldsymbol{G}$ is a valid (identifiable) variance matrix and no further action is required.

5. If $\boldsymbol{R}$ or $\boldsymbol{G} = \boldsymbol{V}_1 \otimes \boldsymbol{V}_2 = \boldsymbol{V}$ then $\boldsymbol{R}$ or $\boldsymbol{G}$ is an over-parameterised (unidentifiable) variance matrix, and it is necessary to determine the scale by fixing one of the scale-dependent parameters in either $\boldsymbol{V}_1$ or $\boldsymbol{V}_2$.

2 Analysis of designed experiments using linear mixed models

In this chapter, we consider the analysis of designed experiments using linear mixed models. In order to determine the appropriate fixed, random and residual components of the models, we follow a procedure similar to that described by Brien & Demetrio (2009), which respects and uses the experimental design. Their ideas give a logical framework for building mixed models. We will work through this procedure to illustrate the process of model determination and fitting using a longitudinal example.

Example 2.1 Queensland fruit fly feeding study

This example and the background information were kindly provided by John Rogers. The data was collected by and is used with the permission of Bugs for Bugs insectary at Mundubbera, owned by Dan Papacek (www.bugsforbugs.com.au), which supplies insect biocontrol agents to the Australian citrus industry as well as providing grower scouting services and working to improve pest management in citrus.

The aim of this trial was to compare the attractiveness of a trial protein bait (coded AY50) against a standard bait (NatFlav) to Queensland fruit fly (QFF). Fruit flies need protein to mature their eggs and are therefore attracted to yeast autolysates (protein baits). These baits are mixed with insecticide and sprayed within orchards - flies are attracted to the protein deposits on leaves, pick up the insecticide and die before they can sting fruit. This trial was undertaken as part of a process to develop improved baits.

Two cages with 25ml of QFF pupae were set up in a constant temperature and humidity environment. On certain days after emergence, the fruit fly were offered two baits. Each time, two petri dishes were put into each cage. Within each cage, each dish had a citrus leaf attached, one with AY50 and the other with NatFlav as protein sources on top of the leaf. The number of flies visiting each of the protein sources was then recorded at 5 minute intervals over a period of 1 hour (12 samples from 0-55 minutes). The number of QFF on each bait is provided for 5 trials conducted 5, 8, 11, 15 and 16 days after emergence.

The data are held in file QFF.txt, which contains factors to describe the structure of the experiment: *Cage* (cage number), *Dish* (dish number within each cage), *FlyAge* (days after emergence), *Minute* (minutes after start of trial); a factor to describe the treatments applied: *Bait* (type of bait); and the response variate (*number*).

We can set up our job, read in and plot this data using the commands

```
# set working directory
setwd("d:/sue/VSNi/AASC/workshop/examples")
getwd()

# load libraries
library(asreml)

# read data
QFF <- asreml.read.table("QFF.txt", header=T)
summary(QFF)

# plot data on natural and log10 scales
xyplot(number ~ Minute | Cage*Bait, data=QFF, groups=FlyAge, type="o", auto.key=T)
xyplot(log10(number) ~ Minute | Cage*Bait, data=QFF, groups=FlyAge, type="o", auto.key=T)
```

The plot produced for the log-scale is shown in Figure 2.1. It seems clear that more flies are attracted to the AY50 bait than to the NatFlav bait, but it is not easy to discern any consistent trends due to fly age, although there seems to be a downwards trend after the first 5 minutes within trials for the NatFlav bait.

□

In the terminology of Brien & Demetrio (2009), this is a two-tiered designed experiment, with two sets of objects: the randomized and unrandomized sets. Note that we regard treatments as randomized onto experimental units in this context. The unrandomized set comprises objects that define the structure of the experiment. Here, the unrandomized objects are the cages, trials within each cage, dishes within each trial and time samples within each dish. The set of unrandomized factors is thus given by (*Cage*, *Dish*, *FlyAge*, *Minute*). The observational units, the smallest units from which a single observation is obtained, are here the sample × dish × trial × cage combinations. The randomized objects are the protein bait treatments, giving a single element set (*Bait*). There are two longitudinal factors here, namely *FlyAge* and *Minute*.

We now construct the intra-tier formulas. First we look at the unrandomized set. Dishes are nested within cages (we do not have information on dish position within cages) (*Cage/Dish*), and for the longitudinal factors, minutes are nested within trials (*FlyAge/Minute*). We would normally cross the longitudinal factors with the other structural factors, giving

$$\textit{(Cage/Dish)*(FlyAge/Minute)} = C + C{:}D + F + F{:}M + C{:}F + C{:}F{:}M + C{:}D{:}F + C{:}D{:}F{:}M,$$

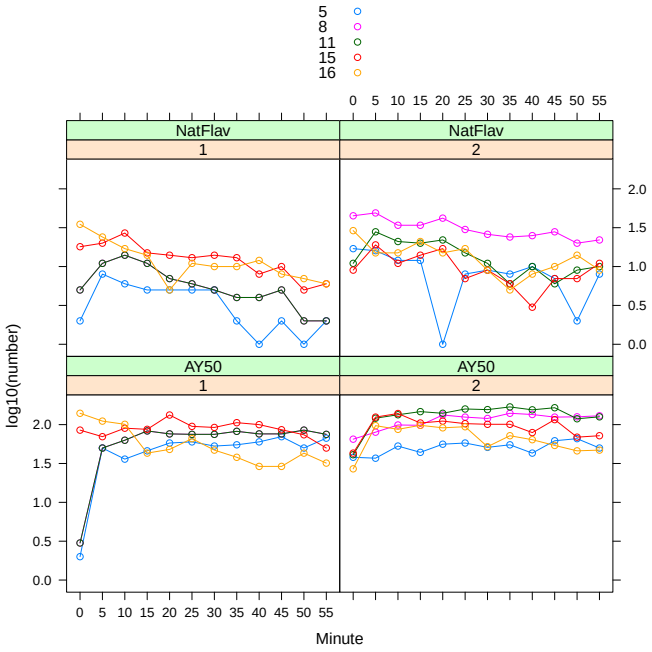

Figure 2.1: Log_{10}(number of flies) landing on each bait in each cage across samples within trials.

using initials for brevity on the right-hand-side in place of full factor names. However, we need to consider whether all of these terms actually make sense. For this experiment the dates of emergence differed between cages, so that trials should be considered as nested within cages. This means that the *FlyAge* and *FlyAge.Minute* structural terms are meaningless and should be removed. In addition, we have no information to enable us to connect dishes within cages across trials (ie. dish 1 in cage 1 in trial 1 may well be unrelated to dish 1 in cage 1 in trial 2), so we also remove the term *Cage:Dish* resulting in the following intra-tier model for the unrandomized factors (known as the intra-tier random (IR) model):

IR: Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge + Cage:Dish:FlyAge:Minute

The final term defines the observation units and so is the residual error term; we will eventually separate this term out from the other random terms.

The intra-tier fixed (IF) model, constructed from the randomized factors, just contains the single factor *IF: Bait*.

We next add inter-tier terms (ie. interactions between factors in the two different sets) that are of interest to the experimenter. In this case, the experimenter is interested in the systematic effects of fly age and exposure time to the attractiveness of the baits, so we supplement the IF model to get the fixed model as

*F: Bait * FlyAge * Minute*

Table 2.1: ANOVA decomposition for Queensland Fruit Fly trial. The residual within each stratum is fitted using the random model term corresponding to the stratum.

Stratum	Model		DF
Source	F/R		
Cage			1
Residual	R	1	
Cage:FlyAge			8
FlyAge	F	4	
Residual	R	4	
Cage:Dish:FlyAge			10
Bait	F	1	
FlyAge:Bait	F	4	
Residual	R	5	
Cage:FlyAge:Minute			110
Minute	F	11	
FlyAge:Minute	F	44	
Residual	R	55	
Cage:Dish:FlyAge:Minute			110
Bait:Minute	F	11	
Bait:FlyAge:Minute	F	44	
Residual	R	55	
Total			239

The fully expanded fixed and random working model formulae are

```
fixed = ~ Bait + FlyAge + Minute + Bait:FlyAge + Bait:Minute +
          FlyAge:Minute + Bait:FlyAge:Minute
random =~ Cage + Cage.FlyAge + Cage.FlyAge.Minute + Cage.Dish.FlyAge +
          Cage.Dish.FlyAge.Minute
```

We can construct an outline ANOVA table for this combined structure, as shown in Table 2.1. There are few residual DF for the overall comparison between baits and their interaction with fly age, but many DF for assessment of the other two-way interactions and the three-way interaction. There are no strata with zero residual DF (which would indicate that some terms must be removed from the random model) and so we can proceed with this model.

We see no reason here to switch terms between the fixed and random models, so our initial (homogeneous) model for analysis has components

```
fixed = ~ Bait + FlyAge + Minute + Bait:FlyAge + Bait:Minute +
          FlyAge:Minute + Bait:FlyAge:Minute
random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge
rcov = ~ Cage:Dish:FlyAge:Minute
```

Recall that in this form, each of the random terms and the residual will be considered as a set of IID effects with a common error variance. The default single section parameterization with θ estimated will be used, with estimates on the ratio scale (see Section 1.3.4). We can fix $\theta = 1$ (and obtain estimates on the component scale) by using the specification

```
rcov = ~ idv(Cage):Dish:FlyAge:Minute
```

We can specify and fit this model using the following commands

```
# sort data frame to match order implied by definition of residual term
QFF <- QFF[order(QFF$Cage,QFF$Dish,QFF$FlyAge,QFF$Minute),]
head(QFF,25)

# simple variance components model
QFFaov <- asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
          Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute,
          random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
          rcov = ~ idv(Cage):Dish:FlyAge:Minute,
          data=QFF)

# print summary of QFFaov object
summary(QFFaov)

# print approximate F tests for fixed terms
wald(QFFaov, denDF="default")
```

We first sort the data frame into the order required by our specification of the residual error term, and then fit the model to create the object *QFFaov*. This object is of the *asreml* class described in more detail in Section 1.4. Function *summary()* applied to the *QFFaov* object produces a summary of the fitted model. Finally, we print approximate F-tests for the fixed model terms. The output is given below.

```
> # simple variance components model
> QFFaov <- asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
+           Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute,
+           random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
+           rcov = ~ idv(Cage):Dish:FlyAge:Minute,
+           data=QFF)

asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), IA32
     LogLik         S2      DF      wall     cpu
     49.1496      1.0000   120  15:10:39     0.0 (4 restrained)
     70.6644      1.0000   120  15:10:39     0.0
...
     82.0698      1.0000   120  15:10:39     0.0
     82.0698      1.0000   120  15:10:39     0.0

Finished on: Sun Nov 04 15:10:39 2012
```

```
LogLikelihood Converged
> # print summary of QFFaov object
> summary(QFFaov)
$call
asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
    Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute, random = ~Cage +
    Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
    rcov = ~idv(Cage):Dish:FlyAge:Minute,
    data = QFF)

$loglik
[1] 82.06976

$nedf
[1] 120

$sigma
[1] 1

$varcomp
                                    gamma   component   std.error   z.ratio
Cage!Cage.var                0.020991231 0.020991231 0.037373824 0.5616560
Cage:FlyAge!Cage.var         0.017517941 0.017517941 0.019463424 0.9000442
Cage:FlyAge:Minute!Cage.var  0.009115796 0.009115796 0.005462513 1.6687916
Cage:Dish:FlyAge!Cage.var    0.013942288 0.013942288 0.010428960 1.3368820
R!variance                   1.000000000 1.000000000          NA        NA
R!Cage.var                   0.030356344 0.030356344 0.005788728 5.2440442
                             constraint
Cage!Cage.var                  Positive
Cage:FlyAge!Cage.var           Positive
Cage:FlyAge:Minute!Cage.var    Positive
Cage:Dish:FlyAge!Cage.var      Positive
R!variance                        Fixed
R!Cage.var                     Positive

attr(,"class")
[1] "summary.asreml"

>
> wald(QFFaov, denDF="default")

asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), IA32
Algebraic ANOVA Denominator DF calculation is not available
Empirical derivatives will be used.

     LogLik          S2      DF      wall     cpu
     82.0698      1.0000   120  15:45:30     0.0

Finished on: Sun Nov 04 15:45:30 2012
```

```
$Wald
                    Df denDF    F.inc           Pr
(Intercept)          1     1 149.0000 5.203763e-02
Bait                 1     5 237.2000 2.094800e-05
FlyAge               4     4   1.4110 3.734882e-01
Minute              11    55   5.1910 1.543861e-05
Bait:FlyAge          4     5   1.1710 4.230328e-01
Bait:Minute         11    55  11.2500 1.498684e-10
FlyAge:Minute       44    55   0.9381 5.836962e-01
Bait:FlyAge:Minute  44    55   1.0150 4.746202e-01

$stratumVariances
NULL
```

The summary shows the log-likelihood and estimated variance parameters. Because we have specified our model to fix $\theta = 1$, the estimates in the gamma and component columns are equal. All of the sources of variation appear substantial. Recall that these random terms have been derived from the physical structure of the experiment, and many define strata within which we estimate treatment terms. We would not therefore consider removing random terms even if their variance components were close to zero, as this would change the set of strata present in the structure.

The *wald()* function prints a table of approximate F-tests for the fixed model terms. In general, the denominator DF (denDF) for these tests are approximate, calculated using the method of Kenward & Roger (1997). For this simple model with balanced data, the residual DF are exactly as we predicted from our outline ANOVA table.

We could examine the results of this fit more closely, but will not do so yet as we have not completed the model selection process. For longitudinal data, the final step of model determination should consider whether there is any evidence of correlation or variance heterogeneity over time.

Here, we first examine plots of residuals over time to see whether they suggest the presence of serial correlation. There are two scales of longitudinal measurement in this example: measurements across 5 minute intervals within trials (equally-spaced) and the repeat of the trials on different days (unequally-spaced). We save and plot the residuals using commands

```
# get residuals
res <- residuals(QFFaov)
fv <- fitted(QFFaov)

# plot residuals to look for suggestion of serial correlation
xyplot(res ~ fv, data=QFF, groups=Bait)
xyplot(res ~ Minute | Cage*Bait, data=QFF, groups=FlyAge, type="a", auto.key=T)
xyplot(res ~ FlyAge | Minute, data=QFF, groups=Cage:Bait, type="a", auto.key=T)
```

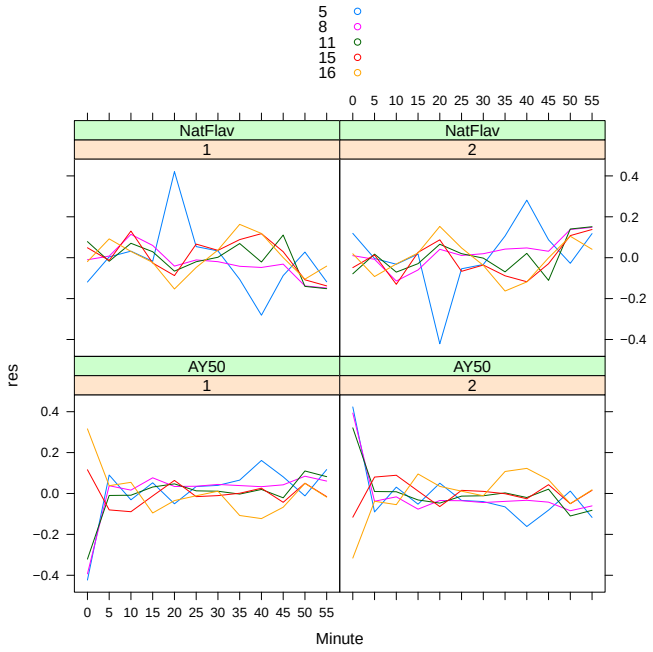

Figure 2.2: Residuals from variance components model for each bait in each cage across samples within trials.

These plots are shown in Figures 2.2 and 2.3. In both cases, the time trends are much smoother than we might expect if no serial correlation were present. This is not surprising for measurements across samples within trials, as these are taken at very close time intervals, and we might expect adjacent measurements to be closely related. It is more surprising when considering successive trials, and we note that with only 4 cage × bait combinations, we have limited information at this level. So we will add serial correlation across samples within trials first, then see if any improvement is made by adding serial correlation across trials.

We add serial correlation using an auto-regressive model of order 1 across samples within trials by changing the *rcov* statement to

```
rcov = ~ idv(Cage):Dish:FlyAge:ar1(Minute)
```

The *ar1* model regards successive factor levels as 1 unit apart, with correlation $\phi^{|i-j|}$ between factor levels i and j. The correlation parameter ϕ is scale-independent. In practice here, the correlation parameter estimates the serial correlation between samples 5 minutes apart. The following commands are used

```
# add serial correlation across samples within trial
QFFar1 <- asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
          Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute,
          random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
          rcov = ~ idv(Cage):Dish:FlyAge:ar1(Minute),
          maxiter=20, data=QFF)
```

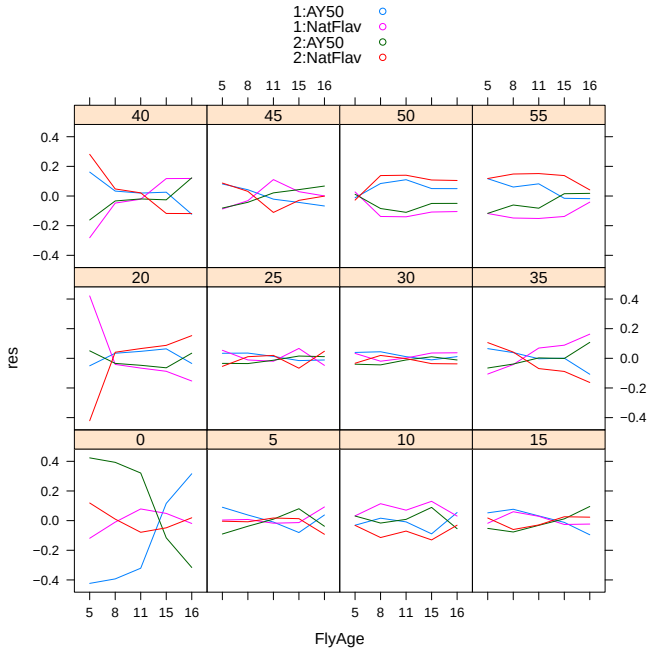

Figure 2.3: Residuals from variance components model for each bait in each cage against fly age at each sample time.

```
# print summary of QFFar1 object
summary(QFFar1)
```

to produce output

```
> # add serial correlation across samples within trial
> QFFar1 <- asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
+           Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute,
+           random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
+           rcov = ~ idv(Cage):Dish:FlyAge:ar1(Minute),
+           maxiter=20, data=QFF)

asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), IA32
     LogLik          S2      DF      wall     cpu
     50.4669      1.0000   120  16:05:25     0.0 (4 restrained)
     73.1235      1.0000   120  16:05:25     0.0
...
     86.6194      1.0000   120  16:05:25     0.0
     86.6195      1.0000   120  16:05:25     0.0

Finished on: Sun Nov 04 16:05:25 2012

LogLikelihood Converged
>
> # print summary of QFFar1 object
```

```
> summary(QFFar1)
$call
asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
    Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute, random = ~Cage +
    Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge, rcov = ~idv(Cage):Dish:FlyAge:ar1(Minute),
    data = QFF, maxiter = 20)

$loglik
[1] 86.61955

$nedf
[1] 120

$sigma
[1] 1

$varcomp
                                  gamma   component   std.error   z.ratio    constraint
Cage!Cage.var                 0.024335493 0.024335493 0.043440655 0.5602009      Positive
Cage:FlyAge!Cage.var          0.023725443 0.023725443 0.022561225 1.0516026      Positive
Cage:FlyAge:Minute!Cage.var   0.006000008 0.006000008 0.004258168 1.4090588      Positive
Cage:Dish:FlyAge!Cage.var     0.005092130 0.005092130 0.009995637 0.5094353      Positive
R!variance                    1.000000000 1.000000000          NA        NA         Fixed
R!Cage.var                    0.042025679 0.042025679 0.010531257 3.9905664      Positive
R!Minute.cor                  0.493064052 0.493064052 0.128224620 3.8453150 Unconstrained

attr(,"class")
[1] "summary.asreml"
```

The estimated serial correlation (at the residual level) between successive samples within trials (0.49) is substantial, as reflected in an increase of 4.55 units in log-likelihood from 82.07 (variance components model) to 86.62 (current model).

We can conduct a formal test of the null hypothesis $H_0 : \phi = 0$ using a likelihood ratio test (LRT), implemented by comparing the REML log-likelihood of the current model (denoted ℓ_A) with that of the model with $\phi = 0$ (denoted ℓ_0). The test statistic takes the form

$$-2(\ell_0 - \ell_A) \, .$$

Under the null hypothesis, this test statistic has a χ^2 distribution on 1 degree of freedom (as one parameter was added), with 95th percentile equal to 3.84. In this case the test statistic has value 9.1 and we therefore reject the null hypothesis and include the auto-regressive process in our model.

Aside: In principle, likelihood ratio tests can be used to compare any two nested random models fitted by REML. In practice, some caution is required. Firstly, this type of LRT cannot be used to compare models with different fixed terms (for details see Welham & Thompson (1997)). Secondly, a modification is required when the null hypothesis coincides with the boundary of

the parameter space for the parameter being tested. We would have encountered this problem here if we had constrained our auto-regressive parameter to remain positive, and this issue commonly occurs when testing variance components that are constrained positive. A mixture of χ^2 *distributions is usually recommended in these situations, but Crainiceanu & Ruppert (2004) showed that this is not always appropriate. This is an area of active research.*

In order to add serial correlation across trials, we need to take into account the fact that the trials are not equally spaced, and so an auto-regressive model (which assumes equal spacing) is not appropriate. Instead, we can use the exponential power model, which is an analogue of the AR1 process for continuous time. This is specified using the *exp()* variance model, using argument *dist* to supply the fly ages

```
# get numeric levels of FlyAge for use in exponential correlation model
age <- unique(QFF$FlyAge)
age

# add serial correlation across trials (allowing for unequal time steps)
QFFar2 <- asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
          Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute,
          random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
          rcov = ~ idv(Cage):Dish:exp(FlyAge,dist=age):ar1(Minute),
          maxiter=25, data=QFF)

# print summary of QFFar2 object
summary(QFFar2)
```

with output

```
> # get numeric levels of FlyAge for use in exponential correlation model
> age <- unique(QFF$FlyAge)
> age
[1] 5  8  11 15 16
Levels: 5 8 11 15 16
>
> # add serial correlation across trials (allowing for unequal time steps)
> QFFar2 <- asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
+           Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute,
+           random = ~ Cage + Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge,
+           rcov = ~ idv(Cage):Dish:exp(FlyAge,dist=age):ar1(Minute),
+           maxiter=25, data=QFF)

asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), IA32
     LogLik         S2      DF      wall     cpu
     51.1382      1.0000   120  16:22:21     0.1 (4 restrained)
     74.6689      1.0000   120  16:22:21     0.1
     86.4466      1.0000   120  16:22:21     0.0
Logliklihood decreased to      72.53 - trying again with reduced updates
     88.1009      1.0000   120  16:22:21     0.1
```

```
...
     99.5577      1.0000   120  16:22:22     0.1
     99.5578      1.0000   120  16:22:22     0.1

Finished on: Sun Nov 04 16:22:22 2012

LogLikelihood Converged
>
> # print summary of QFFar2 object
> summary(QFFar2)
$call
asreml(fixed = log10(number) ~ Bait + FlyAge + Minute + Bait:FlyAge +
    Bait:Minute + FlyAge:Minute + Bait:FlyAge:Minute, random = ~Cage +
    Cage:FlyAge + Cage:FlyAge:Minute + Cage:Dish:FlyAge, rcov = ~idv(Cage):Dish:exp(FlyAge,
    dist = age):ar1(Minute), data = QFF, maxiter = 25)

$loglik
[1] 99.55779

$nedf
[1] 120

$sigma
[1] 1

$varcomp
                                  gamma    component   std.error    z.ratio    constraint
Cage!Cage.var                0.0237824082 0.0237824082 0.043527894  0.5463717      Positive
Cage:FlyAge!Cage.var         0.0223041258 0.0223041258 0.019761961  1.1286393      Positive
Cage:FlyAge:Minute!Cage.var  0.0007790793 0.0007790793 0.001962169  0.3970500      Positive
Cage:Dish:FlyAge!Cage.var    0.0029977084 0.0029977084 0.005627594  0.5326803      Positive
R!variance                   1.0000000000 1.0000000000          NA         NA         Fixed
R!Cage.var                   0.0503655841 0.0503655841 0.011103023  4.5362045      Positive
R!FlyAge.pow                 0.7981012133 0.7981012133 0.059994561 13.3028927 Unconstrained
R!Minute.cor                 0.4095717381 0.4095717381 0.115261586  3.5534106 Unconstrained

attr(,"class")
[1] "summary.asreml"
```

The log-likelihood for this model is much larger again (99.56), with serial correlation between trials on adjacent days estimated as 0.80. Serial correlation (at the residual level) between trials held d days apart is derived as 0.80^d.

Checking the residuals suggests no further issues, and so we examine the approximate F tests for the fixed terms

```
>
> # wald tests for fixed terms in final model
> wald(QFFar2,denDF="default")
```

Table 2.2: Comparison of approximate F-tests for fixed term from simple variance components model and final model with serial correlation in residual term across samples and trials.

	Var. comp. model			Final model		
Term	F	denDF	P	F	denDF	P
Bait	237.20	5	<.001	156.10	12.3	<.001
FlyAge	1.41	4	0.373	1.38	3.7	0.383
Minute	5.19	55	<.001	2.22	17.7	0.065
Bait:FlyAge	1.17	5	0.423	2.68	3.5	0.182
Bait:Minute	11.25	55	<.001	2.47	17.6	0.043
FlyAge:Minute	0.94	55	0.584	1.72	29.4	0.064
Bait:FlyAge:Minute	1.02	55	0.474	1.12	28.0	0.383

```
asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), IA32
Algebraic ANOVA Denominator DF calculation is not available
Empirical derivatives will be used.

     LogLik          S2       DF       wall      cpu
     99.5578       1.0000    120  16:28:16      0.1

Finished on: Sun Nov 04 16:28:17 2012


$Wald
                    Df denDF   F.inc           Pr
(Intercept)          1   1.0 124.400 5.691990e-02
Bait                 1  12.3 156.100 3.081458e-08
FlyAge               4   3.7   1.375 3.825783e-01
Minute              11  17.7   2.217 6.456290e-02
Bait:FlyAge          4   3.5   2.675 1.818683e-01
Bait:Minute         11  17.6   2.469 4.294006e-02
FlyAge:Minute       44  29.4   1.715 6.378235e-02
Bait:FlyAge:Minute  44  28.0   1.118 3.831789e-01

$stratumVariances
NULL
```

It is instructive to compare the denominator DF for these approximate F tests with those obtained from the simple variance components model. This comparison is made in Table 2.2. The presence of positive serial correlation across samples within trials and across trials effectively reduces the replication according to the strength of the serial correlation, resulting in fewer denominator DF.

These results indicate large differences between the two baits, and that this pattern changes across time within trials, but there appears to be no effect of fly age on the response. To complete our analysis, we produce predictions. The model we would use for prediction here is

then *Bait*Minute*. In this balanced design, there is no need to re-fit the model before making predictions. We form and view predictions using the commands

```
> # get predictions from final model
> # (for balanced data, no need to refine fixed model before predict)
> QFFar2.pv <- predict(QFFar2,classify=c("Bait:Minute"))

asreml 3.0-1 (20 July 2012), Library: 3.0hj (15 November 2011), IA32
     LogLik          S2      DF      wall     cpu
     99.5578      1.0000   120  17:02:38     0.1
     99.5578      1.0000   120  17:02:38     0.0
     99.5578      1.0000   120  17:02:38     0.1
     99.5578      1.0000   120  17:02:38     0.1

Finished on: Sun Nov 04 17:02:38 2012

LogLikelihood Converged
> QFFar2.pv$predictions
$pvals

Notes:
- The predictions are obtained by averaging across the hypertable
  calculated from model terms constructed solely from factors in
  the averaging and classify sets.
- Use "average" to move ignored factors into the averaging set.

- The SIMPLE averaging set:  FlyAge
- The ignored set:  Cage Dish

       Bait Minute predicted.value standard.error est.status
1      AY50      0       1.3401132      0.1622155  Estimable
2      AY50      5       1.8625090      0.1622155  Estimable
3      AY50     10       1.9043097      0.1622155  Estimable
4      AY50     15       1.8888313      0.1622155  Estimable
5      AY50     20       1.9352680      0.1622155  Estimable
6      AY50     25       1.9373242      0.1622155  Estimable
7      AY50     30       1.8810485      0.1622155  Estimable
8      AY50     35       1.9149111      0.1622155  Estimable
9      AY50     40       1.8660117      0.1622155  Estimable
10     AY50     45       1.8907259      0.1622155  Estimable
11     AY50     50       1.8557576      0.1622155  Estimable
12     AY50     55       1.8223032      0.1622155  Estimable
13 NatFlav      0       1.0840005      0.1622155  Estimable
14 NatFlav      5       1.2463436      0.1622155  Estimable
15 NatFlav     10       1.1882583      0.1622155  Estimable
16 NatFlav     15       1.1484012      0.1622155  Estimable
17 NatFlav     20       0.9606476      0.1622155  Estimable
18 NatFlav     25       1.0042458      0.1622155  Estimable
19 NatFlav     30       0.9562132      0.1622155  Estimable
```

```
20 NatFlav     35       0.8157667      0.1622155  Estimable
21 NatFlav     40       0.7964542      0.1622155  Estimable
22 NatFlav     45       0.8517565      0.1622155  Estimable
23 NatFlav     50       0.6693657      0.1622155  Estimable
24 NatFlav     55       0.7700540      0.1622155  Estimable

$avsed
  overall
0.1512037
```

In order to plot the predictions, we need to save (and in this case reformat) the predictions, their SE and the factors indexing the set

```
# extract predictions
QFFar2.pred <- QFFar2.pv$predictions$pvals$predicted.value
QFFar2.pse <- QFFar2.pv$predictions$pvals$standard.error
QFFar2.pB <- QFFar2.pv$predictions$pvals$Bait
QFFar2.pM <- QFFar2.pv$predictions$pvals$Minute
# derive lower and upper approx 95% CI
QFFar2.low <- QFFar2.pred - 2*QFFar2.pse
QFFar2.upp <- QFFar2.pred + 2*QFFar2.pse
QFFar2.index <- c(1:length(QFFar2.pred))

# make data frame containing predictions
QFFar2.predict <- data.frame(id=QFFar2.index, pred=QFFar2.pred,se=QFFar2.pse,low=QFFar2.low,
                             upp=QFFar2.upp,Bait=QFFar2.pB,Minute=QFFar2.pM)

# need to reshape (stack) data frame to plot simply with CI
long <- reshape(QFFar2.predict, varying = list(c(2,4,5)), direction = "long")
long$time <- as.factor(long$time)

# plot predictions with CI on natural and back-transformed scales
pset<- simpleTheme(col=rep(c("red","blue"),each=3),lty=rep(c(0,1,1),2))
xyplot(pred ~ Minute , group=Bait:time, data=long, auto.key=T, type="o",
  par.settings=pset)
xyplot(10^pred ~ Minute , group=Bait:time, data=long, auto.key=T, type="o",
  par.settings=pset)
```

The resulting plot on the natural scale (with predictions and CI back-transformed) is shown in Figure 2.4. It is clear that there is an increase between 0 and 5 minutes. At time zero, there appear to be similar numbers of flies on the two baits, but the numbers increase much more at the 5 minute sample for the AY50 bait than for the NatFlav bait. There is a clear decrease over samples within trial for the NatFlav bait, but no clear pattern for the AY50 bait. As the response after 5 minutes looks approximately linear on the log-scale, we could construct covariates to investigate this further, but will leave this example here for now.

In this chapter, we have followed through the analysis of one particular designed experiment. In general, model determination for designed experiments should be largely driven by the random-

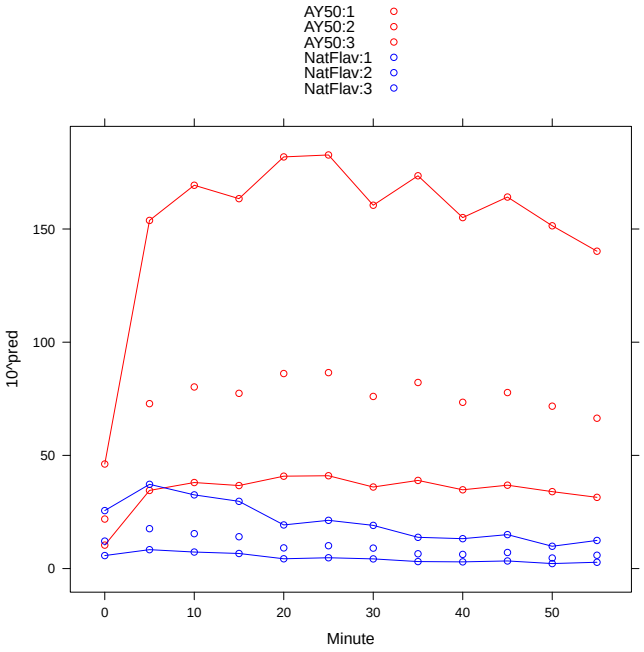

Figure 2.4: Prediction from final model with approximate 95% CI back-transformed onto natural scale.

ization structure and aim to preserve the experimental strata. This approach is somewhat at odds with the so-called variance modelling approach, which seeks to find the most parsimonious structure that adequately describes the variation present. The variance modelling approach allows small (non-significant) variance components to be discarded from the model, which may be inappropriate in the context of designed experiments, as it changes the underlying strata and their associated DF.

3 Using relationship matrices for estimation of genetic effects

3.1 Introduction

ASReml has a range of facilities for undertaking genetic analyses, in particular fitting models where information on pedigrees is incorporated into the analysis. In this chapter we will illustrate some of these facilities using a data-set kindly provided by Dr. Canhong Cheng from the New Zealand Institute for Plant and Food Research. The aim of the analysis was to estimate genetic parameters for so-called general combining ability (GCA) and specific combining ability for a range of traits in kiwifruit. General combining ability is synonymous with what is referred to as additive genetic effects, while specific combining ability is more generally referred to as dominance genetic effects. We will not attempt a comprehensive account of the underlying genetic models; the reader is referred to texts such as Mrode (1995); Lynch & Walsh (1998). Most of the details concerning the approach we use here can be found in Oakey et al. (2006, 2007). Additional concepts will be introduced and briefly described as they arise.

One of the more common applications of our approaches is in the analysis of hybrid crops such as maize and sorghum. The phenomena of hybrid vigour sometimes called heterosis has been exploited in these crops through the deployment of F1 hybrids produced by crossing genetically diverse inbred parent lines. The F1 progeny yield better than either of the inbred parents due to the impact of heterosis. Because the hybrids are heterozygous at loci that are polymorphic between the parents, additive, dominance and epistatic components of genetic variance contribute to differences in traits in so-called hybrid individuals compared with their inbred parents where the dominance is absent due to the lack of heterozygosity.

Typically hybrid breeding programs, such as the sorghum breeding programme at DAFF (formerly QDPI) identify elite cultivars by assessing the performance of large numbers of different combinations of inbred parent lines in multi-environment trials. For many years it has been common practice to assess average variety performance by conducting a series of breeding trials in multiple locations. Multi-environment trials (METs) have been analysed using the mixed model approach outlined in Smith et al. (2001). These approaches assess variety performance by considering all varieties as independent genetic lines without allowing for the fact that they may or may not be parentally related. Allowing for parental relatedness is of value particularly

in breeding hybrid crops such as sorghum, where the parents contain the information required for both the development of new inbred parents and the identification of superior hybrid combinations.

It has recently been suggested that models that allow for additive and non-additive genetic effects should be considered. This has been achieved by partitioning the genetic variance component via the use of a relationship matrix $\boldsymbol{A}$ as in Oakey et al. (2006, 2007). The accuracy of the additive effects can be improved by the addition of a dominance effect (Maki-Tanila (2007)). Dominance genetic effects are variations due to the interaction of alleles at the same locus. Oakey et al (2007) discuss partitioning the genetic effects into additive, dominance and residual genetic parts and demonstrate the application of these ideas with a multi-environment trial data-set. However, Oakey et al. (2007) raised concerns with the implementation of these models for large data-sets or for data-sets with a large pedigree. Oakey et al. (2007) present a less computationally demanding approach but note that there may still be limitations to the implementation as the matrices involved need to be inverted and these matrices are often relatively dense.

In this course we present an alternative approach for computing the dominance relationship matrix via a Monte Carlo simulation approach but recognise that there remain computational challenges for larger data-sets. Full details of the approach are available in an unpublished technical report written by Ari Verbyla. The **ASReml-R** implementation of this is due to David Butler.

3.2 IIllustrative example

3.2.1 Genetic material

The genetic material used in this experiment was derived from a disconnected mating factorial, with 2×6 and 2×2 sub-factorials and a total of 16 full-sib families of kiwifruit. Table 3.1 presents a summary of this mating design where males 1 to 6 are non-red, males 7 and 8 are red, females 1 and 2 are red and females 3 and 4 are non-red. Females 1 and 2 and males 7 and 8 had high dry matter but short storage life, while females 3 and 4 and males 1 to 6 had a long storage life and moderate dry matter. Full-sib families between red parents or between non-red parents were not produced.

	Male							
Female	Dad1	Dad2	Dad3	Dad4	Dad5	Dad6	Dad7	Dad8
Mum1	40	50	42	47	42	48	0	0
Mum2	45	42	51	49	38	49	0	0
Mum3	0	0	0	0	0	0	46	51
Mum4	0	0	0	0	0	0	44	43

Table 3.1: Summary of numbers of (female) vines with data for each full-sib family

3.2.2 Phenotypic data

A trial was established in 2007 at the Te Puke Research Centre to progeny test the above set of full-sib families with the aim of identifying superior parents for use in the kiwifruit breeding programme, particularly male parents as kiwifruit is dioecious. The trial was set out in a completely randomized design with 10 plantings of each of the 16 full-sibs families. Vines were planted in groups of 10 along an orchard row with a spacing of 0.4m between each vine. The vines were trained on a T-bar trellis system. Table 3.2 presents a summary of the allocation of the full-sib families and the numbers of female vines per planting unit (ie a combination of the un-randomised factors `Row` and `Bay`.

	Row							
Bay	01	02	03	04	05	06	07	08
00	A6;1	A1;3	A5;3	B4;2	B5;1		D8;2	
01	A2;3	B1;5	A3;2	A5;7	A6;7	B2;6	C7;3	C8;6
02	B3;5	A4;6	A1;2	A2;6	A5;3	B4;5	C8;5	D8;5
03	B1;6	A5;4	A2;5	B6;2	B3;6	A3;6	D7;3	C7;1
04	B2;3	B5;6	B4;3	B3;6	A5;5	A2;4	D7;6	D8;4
05	A5;4	A6;5	B3;2	A1;3	B5;5	A4;4	C7;4	C8;5
06	B5;5	A2;6	A6;8	B2;3	B3;6	A5;2	C8;6	C7;7
07	B4;6	A5;3	B5;6	A6;6	A3;2	A1;4	D8;3	D7;5
08	A1;2	B4;7	B6;2	A4;2	B1;7	A5;6	C8;6	C7;5
09	B6;4	B1;3	A5;5	A3;5	A4;7	A6;6	D7;5	C8;6
10	A3;4	A4;7	B1;4	B5;3	B6;7	A1;3	D8;4	D7;4
11	A4;6	B2;5	A3;5	B1;5	A1;6	B5;2	C7;7	D8;7
12	A3;5	A1;6	B5;2	A4;5	B2;6	B3;5	D8;4	D7;5
13	A6;5	B3;8	A1;6	B6;9	B4;5	B1;5	C8;5	C7;6
14	B2;5	A3;3	B3;6	A2;8	A6;5	B6;5	C7;4	D8;2
15	B5;5	B6;6	B4;6	A3;5	A2;2	A4;3	D7;6	C8;5
16	B6;6	B1;7	B2;2	A1;5	B3;5	B4;5	D8;7	D7;3
17	B4;5	B2;6	A4;6	B1;3	A6;4	A2;7	C8;5	D8;5
18	A2;5	B6;6	A6;1	B4;5	B5;3	B2;4	D7;7	C7;6
19	B3;1	A4;1	A2;3	B2;2	B6;1	A3;3	C7;1	C8;1
20	B3;1		A2;1		B6;1	A3;2	C7;2	C8;1

Table 3.2: Summary of field layout of full-sib families and numbers of female vines for planting group

3.3 Results

Undertaking a genetic analysis using information on the relatedness of individuals requires a pedigree file and an associated file containing the phenotypic data, which is linked to the pedigree file, usually through the column providing the coding for individuals (ie. vines). This column ***must*** correspond to a column in the data-file and be called the same name. Individuals which are not present in the data, can and usually are included in the pedigree file, for example, parents, grand-parents and so on, however it is necessary to include all individuals in the data-file in the

pedigree file. Chapter 5 of the **ASReml-R** manual describes the syntax of the commands which are used to create the additive relationship matrix, ie. $\boldsymbol{A}$ and how to link this into the analysis. We use the information in this chapter but also demonstrate how to incorporate dominance into the analysis following Oakey et al. (2007).

3.3.1 Creating the additive relationship matrix

The pedigree file defines the genetic relationships of the individuals and in its simplest form contains three fields, namely the individual, the female parent and the male parent. If either or both parent(s) is(are) unknown (eg. in the case of open-pollinated plants or founders) then we simply place an *NA* in the field. The **ASReml-R** manual does state that you can use a zero for an unknown parent but we find this a little confusing and prefer to use a *NA*. Our pedigree file contains a total of 772 individuals, 45 ancestors and the remaining individuals being vines which occur in the data file. Five founders have unknown male and female parents, while 9 individuals have unknown male parents. There are 727 individuals with data.

To create $\boldsymbol{A}^{-1}$ we use the following commands:

```
kiwiped.df <- read.table('kiwiped.csv',sep=',',header=T,as.is=T)
kiwiped.df$female[kiwiped.df$female=='0'] <- NA
kiwiped.df$male[kiwiped.df$male=='0'] <- NA
kiwiped.df <- kiwiped.df[,c(4,2,3)]
names(kiwiped.df) <-c('Vine','Female','Male')
kiwi.ainv <- asreml.Ainverse(kiwiped.df)
```

Note that the input file has column names which were not the same as the column names in the data file, necessitating a change. The object *kiwi.ainv* which is returned from **asreml.Ainverse** is a list which has the following elements:

```
> names(kiwi.ainv)
[1] "pedigree"   "ginv"       "inbreeding" "det"        "ifault"
```

The most frequently used components of the list are *pedigree, ginv* and *inbreeding*. The first one is a data frame version of the original pedigree file which has been sorted and tidied up as necessary. The second component is a data frame containing a sparse matrix representation of $\boldsymbol{A}^{-1}$, which is generally used in the call to **ASReml-R**. The remaining component of most interest is the *inbreeding* which contains the inbreeding coefficient for each individual in the pedigree. There is only one in-bred individual in the pedigree:

```
> table(kiwi.ainv[['inbreeding']])

    0 0.125
  771     1
> subset(kiwiped.df,Vine==names(kiwi.ainv$inbreeding[kiwi.ainv$inbreeding==.125]))
        Vine  Female    Male
15 52-13-22d CK51_06 CK51_11
```

and examination of the pedigree file shows how this has occurred, viz:

```
> head(kiwiped.df,12)
      Vine Female Male
1     CK01   <NA> <NA>
2     CK15   <NA> <NA>
3     CK51   <NA> <NA>
4     CK65   <NA> <NA>
5     CK64   <NA> <NA>
6  CK65_09   CK65 <NA>
7  CK64_18   CK64 <NA>
8  CK15_03   CK15 <NA>
9  CK15_01   CK15 <NA>
10 CK51_06   CK51 <NA>
11 CK51_09   CK51 <NA>
12 CK51_11   CK51 <NA>
```

since the parents of this vine share a common mother.

3.3.2 Creating the dominance relationship matrix

The elements (i, j) of the genetic covariance matrices (including $\boldsymbol{A}$ and $\boldsymbol{D}$, the dominance covariance matrix, can be approximated by a Monte Carlo simulation approach, where the detailed identity coefficients, Δ_s, (Gillois (1964); Harris (1964)) are estimated for all pairs of individuals. These specify, given the four genes of individuals i and j at a single locus, the probabilities of partitioning these alleles into the $S = 1, 2, \ldots 15$ (distinguishing between parental type) identical by descent (IBD) groups, and together with the inbreeding coefficients (F_i) form the basis of the various relationship matrices, including $\boldsymbol{A}$ and dominance $\boldsymbol{D}$.

Given a pedigree file, the package **asreml.monte** provides estimates of the Δ_{ijs} for each pair of individuals using a simulation approach in which the pairwise coefficients Δ_{ijs} are estimated as

$$\hat{\Delta}_{ijs} = \delta_{ijs}/N,$$

where the genes of all individuals are examined pairwise (i, j), and counts of events δ_{ijs} contributing to identity state S are accumulated. The inbreeding coefficients are estimated as

$$F_i = \mathcal{F}_\rangle/\mathcal{N}$$

where $\mathcal{F}_\rangle$ is incremented if the genes of an individual i are identical. The elements of the genetic relationship matrices are then derived from these quantities using the results in say chapter 7 of Lynch & Walsh (1998).

The following illustrates the use of `asreml.monte` for these data, to compute $\boldsymbol{D}$:

```
kiwi.mon <- asreml.monte(kiwiped.df,nsim=10000)
kiwi.mon.A <- kiwi.mon$A
kiwi.mon.D <- kiwi.mon$D
```

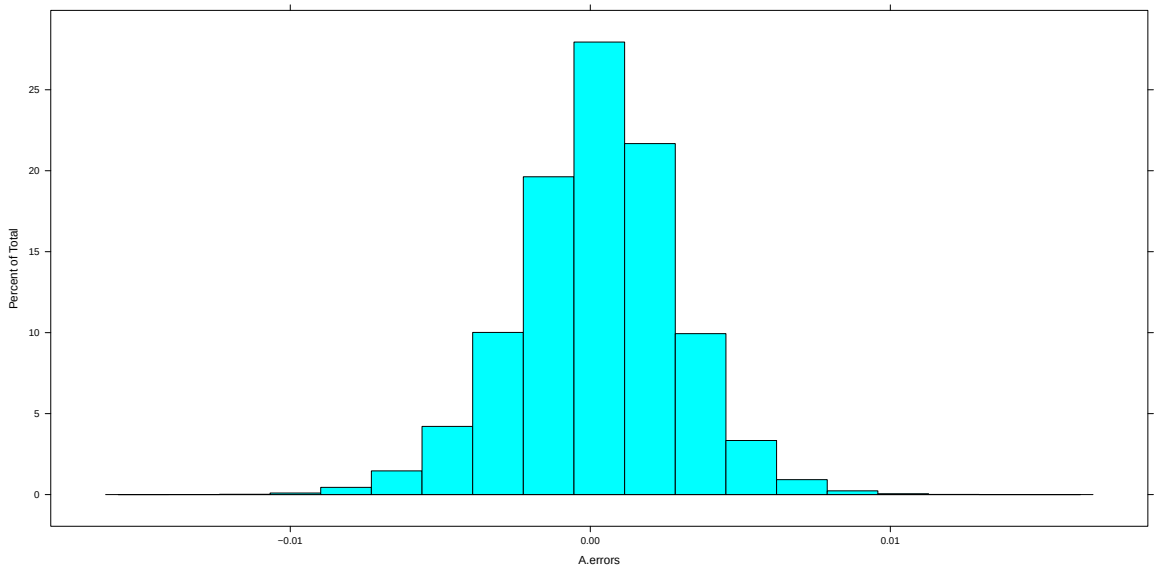

Figure 3.1: Histogram of the difference between simulated and "observed" $\boldsymbol{A}$.

```
kiwi.real.A <- solve(asreml.sparse2mat(kiwi.ainv[['ginv']]))
mn.cop <- mean(kiwi.real.A[lower.tri(kiwi.real.A)])
A.errors <-
   as.vector(kiwi.mon.A - kiwi.real.A)[lower.tri(kiwi.mon.A - kiwi.real.A)]
histogram(A.errors)
```

Out of interest we have also used the monte carlo estimate of $\boldsymbol{A}$ more of a check of the reliability of the choice of N. The histogram of deviations between the monte carlo $\boldsymbol{A}$ and the (true) ancestral based $\boldsymbol{A}$ is presented in figure 3.1.

Any (known, scaled) variance matrix computed from an external source can be included in the analysis by use of the *ginv* option of **asreml**. The inverse can be provided in three forms:

1. the structure returned from **asreml.Ainverse**, which is the data frame representation of $\boldsymbol{A}^{-1}$ in sparse form

2. the complete lower triangle stored as a vector, `NAs` are ignored

3. a matrix object and `NAs` are ignored

We prefer use of (1) to avoid any confusion and for consistency with what is provided by **asreml.Ainverse**. The commands to produce this from the output from *asreml.monte* are as follows:

```
kiwi.mon.iD <- solve(kiwi.mon.D)
kiwi.dinv <-
    data.frame(Column=as.vector(row(kiwi.mon.iD)[!lower.tri(kiwi.mon.iD)]),
```

```
                Row=as.vector(col(kiwi.mon.iD)[!lower.tri(kiwi.mon.iD)]),
                value=as.vector(kiwi.mon.iD[!lower.tri(kiwi.mon.iD)]))
nrow(kiwi.dinv)/772/773
kiwi.dinv <- subset(kiwi.dinv,value!=0)
kiwi.ginv <- kiwi.ainv[['ginv']]
```

At this point we investigate the structure of the dominance relationship matrix. Below is a sub-matrix of $\boldsymbol{D}$ for one vine from each of the 16 full-sib families:

```
> tmp <- with(kiwi.df,split(as.character(Vine),ID))
> tmp1 <- sapply(tmp,function(x) x[1])
> Db <- kiwi.mon.D[tmp1,tmp1]
> dimnames(Db) <- list(names(tmp1),names(tmp1))
> Db
       A1     A2     A3     A4     A5     A6     B1     B2     B3     B4     B5
A1 1.0000 0.1233 0.1245 0.1228 0.1243 0.0000 0.1517 0.0796 0.0769 0.0779 0.0779
A2 0.1233 1.0000 0.1299 0.1215 0.1261 0.0000 0.0785 0.1545 0.0785 0.0752 0.0780
A3 0.1245 0.1299 1.0000 0.1228 0.1215 0.0000 0.0790 0.0790 0.1548 0.0805 0.0792
A4 0.1228 0.1215 0.1228 1.0000 0.1273 0.0000 0.0729 0.0788 0.0745 0.1561 0.0774
A5 0.1243 0.1261 0.1215 0.1273 1.0000 0.0000 0.0770 0.0771 0.0778 0.0768 0.1577
A6 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000
B1 0.1517 0.0785 0.0790 0.0729 0.0770 0.0000 1.0000 0.1387 0.1401 0.1386 0.1380
B2 0.0796 0.1545 0.0790 0.0788 0.0771 0.0000 0.1387 1.0000 0.1426 0.1399 0.1425
B3 0.0769 0.0785 0.1548 0.0745 0.0778 0.0000 0.1401 0.1426 1.0000 0.1376 0.1399
B4 0.0779 0.0752 0.0805 0.1561 0.0768 0.0000 0.1386 0.1399 0.1376 1.0000 0.1412
B5 0.0779 0.0780 0.0792 0.0774 0.1577 0.0000 0.1380 0.1425 0.1399 0.1412 1.0000
B6 0.0000 0.0000 0.0000 0.0000 0.0000 0.1529 0.0000 0.0000 0.0000 0.0000 0.0000
C7 0.0109 0.0121 0.0113 0.0126 0.0116 0.0000 0.0111 0.0119 0.0115 0.0107 0.0105
C8 0.0108 0.0132 0.0113 0.0120 0.0106 0.0000 0.0287 0.0322 0.0289 0.0265 0.0294
D7 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
D8 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
       B6     C7     C8     D7     D8
A1 0.0000 0.0109 0.0108 0.0000 0.0000
A2 0.0000 0.0121 0.0132 0.0000 0.0000
A3 0.0000 0.0113 0.0113 0.0000 0.0000
A4 0.0000 0.0126 0.0120 0.0000 0.0000
A5 0.0000 0.0116 0.0106 0.0000 0.0000
A6 0.1529 0.0000 0.0000 0.0000 0.0000
B1 0.0000 0.0111 0.0287 0.0000 0.0000
B2 0.0000 0.0119 0.0322 0.0000 0.0000
B3 0.0000 0.0115 0.0289 0.0000 0.0000
B4 0.0000 0.0107 0.0265 0.0000 0.0000
B5 0.0000 0.0105 0.0294 0.0000 0.0000
B6 1.0000 0.0000 0.0000 0.0000 0.0000
C7 0.0000 1.0000 0.0587 0.0000 0.0000
C8 0.0000 0.0587 1.0000 0.0000 0.0000
D7 0.0000 0.0000 0.0000 1.0000 0.0635
D8 0.0000 0.0000 0.0000 0.0635 1.0000
```

This reflects the nature of the genetic design (ie. disconnected factorial with essentially non-inbred parents). A portion of $\boldsymbol{D}$ for vines from the same family is:

```
> Dw <- kiwi.mon.D[tmp[[1]],tmp[[1]]]
> Dw[1:6,1:6]
           12E-01-08b 12E-01-08g 12E-02-00a 12E-02-00b 12E-02-00c 12E-02-12a
12E-01-08b     1.0000     0.2473     0.2513     0.2476     0.2515     0.2480
12E-01-08g     0.2473     1.0000     0.2524     0.2504     0.2588     0.2559
12E-02-00a     0.2513     0.2524     1.0000     0.2516     0.2500     0.2522
12E-02-00b     0.2476     0.2504     0.2516     1.0000     0.2482     0.2460
12E-02-00c     0.2515     0.2588     0.2500     0.2482     1.0000     0.2525
12E-02-12a     0.2480     0.2559     0.2522     0.2460     0.2525     1.0000
```

3.3.3 Model development and data manipulation

The phenotypic data is provided in a separate file and we simply create a data frame which contains this information as follows:

```
kiwi.df <- read.table('kiwidata.csv',sep=',',header=T)
head(kiwi.df)
with(kiwi.df,table(Mother,Father))
```

There are a number of problems which we need to resolve. Firstly, we note that the fields containing the parents of the vines are labelled differently to what is in the pedigree file. Additionally the coding of the parents within each of the fields is not the same as the coding used within the pedigree file. Although this is not a major problem, it could present issues if these fields are used in any analysis which includes information on the relatedness of the vines derived from the pedigree file. It is therefore suggested these fields are both named in a consistent manner and contain the same codes.

Another issue which we need to deal with is to create the appropriate set of experimental design factors which allow for an analysis which respects the restricted randomisation processes used in this experiment. We follow the approach described earlier. The experiment was described as a completely randomised design, though this is not strictly the case. The coding of the vines in this experiment (and other experiments) adopts the following convention, with a field containing the orchard block, the orchard row, the orchard bay within a row and the orchard plant within a bay. The vine field is coded as `ooo-rr-bbp`. This experiment is a two-tier experiment. The set (tier) of unrandomised factors is given by (*Block, Row, Bay, Plant*) and the randomised tier is (*ID*), where

`Block:` a factor with two levels assigned a value of 1 for orchard rows 1 to 6 else 2.

`Row:` a factor denoting the orchard row

`Bay:` a factor denoting the orchard bay (within a row)

`Plant:` a factor denoting the plant within orchard bay within orchard row.

The observational unit is the *Vine*. The intra-tier random model formula is formed by consideration of the intrinsic relationship between the factors within the unrandomised tier and is given by

```
Block/(Row:Bay)/Plant = Block + Block:Row:Bay + Block:Row:Bay:Plant
```

where the last term is equivalent to *Vine* and *units*. The nesting of *Bay* within *Row* is interesting. This is how the randomisation was performed but perhaps a more appropriate intra-tier model formula should be

```
Block/Row/Bay/Plant = Block + Block:Row + Block:Row:Bay + Block:Row:Bay:Plant
```

We will use this but also add *Bay* in our subsequent modelling.

Creating the design factors described above is tedious and can be done in a variety of ways. We used **R** and the commands are:

```
temp <- strsplit(as.character(kiwi.df$Vine),split='-')
unique(sapply(temp,function(x) x[[1]])) #[1] "12E"
table(sapply(temp,function(x) x[[2]]))
# 01  02  03  04  05  06  07  08
# 87 103  80  92  94  87  95  89
kiwi.df$Row <- factor(sapply(temp,function(x) x[[2]]))

tmp <- sapply(temp,function(x) substring(x[[3]],1,2))
kiwi.df$Bay <- factor(tmp)
tmp <- sapply(temp,function(x) substring(x[[3]],3,3))
kiwi.df$Plant <- factor(tmp)
table(sapply(temp,function(x) x[[2]]))
kiwi.df$Block <- factor(is.element(sapply(temp,function(x) x[[2]]),c('07','08')))
with(kiwi.df,table(Row,Block))  #$ (formatting issue in emacs!)
```

3.3.4 Model fitting

The main aim of the experiment is to predict breeding values (ie. additive effects, or GCA) for males, but to also include dominance effects (ie. SCA) and we will illustrate this using the fruit number (*FN11*)and fruit weight (*FW11*). There were additional traits in the data file. Some of these traits are either binary or grouped failure time with about 33% of the vines being right censored (ie still unaffected). There are facilities in **ASReml** for fitting generalised linear mixed models to data with non-gaussian errors. **ASReml-R** uses the PQL approach (Breslow & Clayton, 1993) and section 3.12 of the **ASReml-R** manual provides a summary of our current views on the utility of this approach for analysing binary data or grouped failure time data with heavy censoring and complex variance structures. For today we therefore focus on the traits which are more likely to be approximately gaussian.

Before proceeding we make the observation that like most field trials with outbreeding species, such as kiwifruit, there is no true replication, in that vines within a full-sib family cannot be regarded as replicates. The residual genetic variance is therefore completely confounded with other sources of variation which are not accounted for by the terms included in the following models.

To undertake the modelling using both additive and dominance genetic effects we also need to create a copy of the factor which identifies the individual vine to associate the dominance *giv* matrix with. This is achieved using the following commands:

```
require(asreml)
head(cbind(dimnames(kiwi.mon.D)[[1]],attr(kiwi.ginv,'rowNames')))
tail(cbind(dimnames(kiwi.mon.D)[[1]],attr(kiwi.ginv,'rowNames')))
sum(dimnames(kiwi.mon.D)[[1]]==attr(kiwi.ginv,'rowNames'))
temp <- as.character(kiwi.df$Vine)
attr(kiwi.dinv,'rowNames') <- attr(kiwi.ginv,'rowNames')
kiwi.df$DVine <- factor(temp,levels=attr(kiwi.dinv,'rowNames'))
```

where we take care to ensure that the factor *DVine* has the same levels as *Vine*. This will helps producing summaries of the analyses.

Figure 3.2 presents a pairs plot of the traits which we focus on and some additional two traits of interest, namely dry matter (*DM*) and soluble solids (*SSC*). We also include the transformed fruit number in this plot. There appears to be little relationship between fruit number per vine and fruit weight per fruit which is surprising. Fruit number per vine is highly skewed.

In all of the following analyses we maintain the same base-line model. This model is the linear mixed model surrogate for the randomisation based analysis which include terms in the intra-tier model formula. Generally these terms are fitted as random effects, however we here choose to fit *Block* as a fixed effect. There are (at least) two reasons for this. Firstly, given there is complete confounding of some of the genetic effects with *Block* we do not wish to recover information between blocks. Secondly, as the default constraints for variance components in **ASReml** is *positive*, then often strata are lost if the REML estimate of a variance component associated with a random term is fixed at zero.

Since we are interested in selection and incorporation of pedigree information then all terms associated with genetic effects are included as random effects. The following presents the output from our initial analysis of fruit number per vine:

```
> kiwi.asr <- asreml(sqfn~1+Block,random=~Row + Bay + Row:Bay +
+                     ped(Vine) + ped(DVine),data=kiwi.df,na.method.X='include',
+                     rcov=~idv(units),ginverse=list(Vine=kiwi.ginv,DVine=kiwi.dinv))

asreml 3.0-1 (24 October 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik         S2      DF      wall     cpu
  -3545.8635      1.0000   669  17:07:51     1.4
```

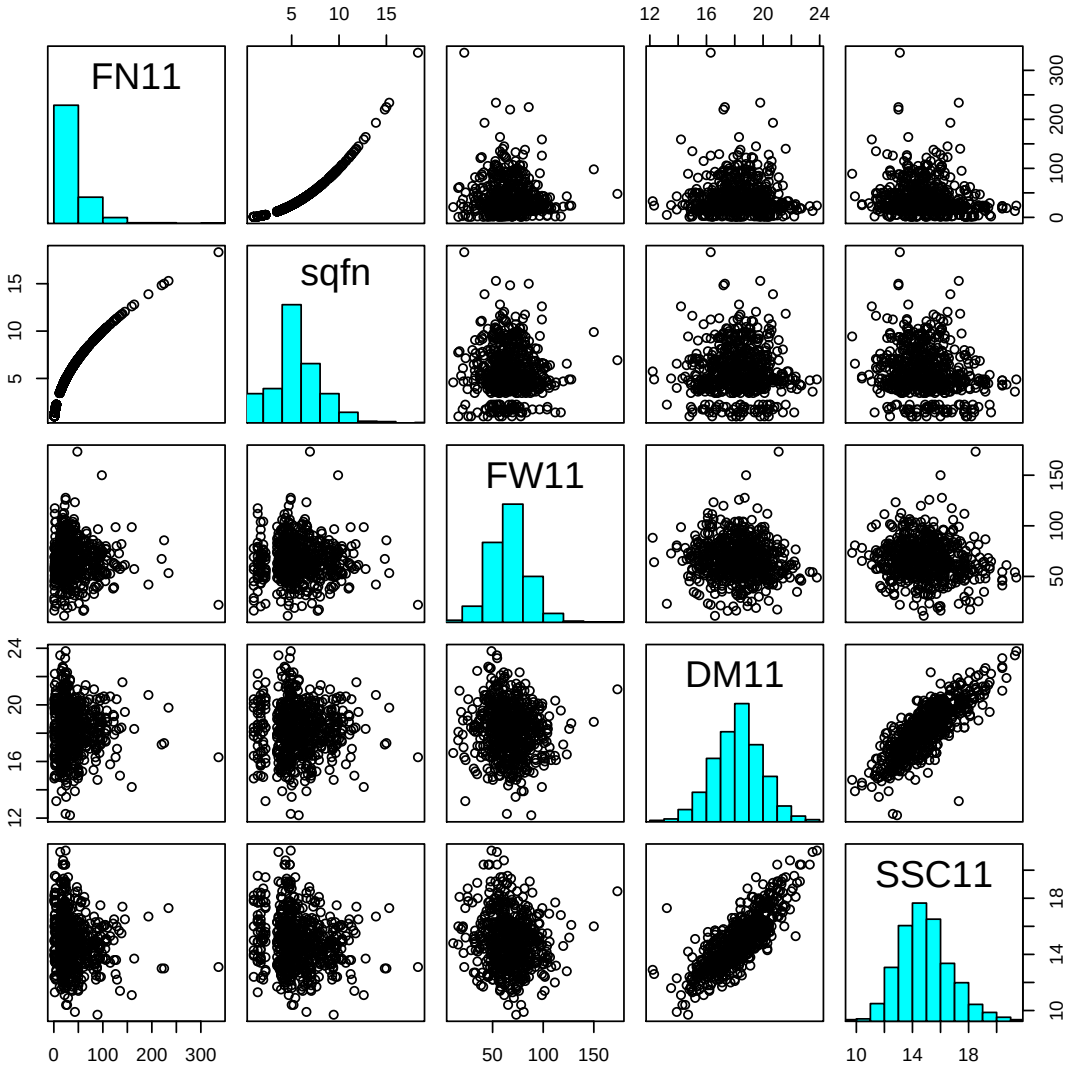

Figure 3.2: Pairs plot of five traits for the kiwifruit example.

```
  -2794.3802      1.0000   669  17:07:52     0.8
  -1964.2795      1.0000   669  17:07:53     1.0
  -1286.4346      1.0000   669  17:07:54     0.9
  -1026.6994      1.0000   669  17:07:55     0.8 (1 restrained)
   -961.0271      1.0000   669  17:07:55     0.8 (1 restrained)
   -942.0140      1.0000   669  17:07:56     0.8 (1 restrained)
   -938.2845      1.0000   669  17:07:57     0.8 (1 restrained)
   -937.8095      1.0000   669  17:07:58     0.8
   -937.7610      1.0000   669  17:07:59     0.8
   -937.7605      1.0000   669  17:08:00     0.8
   -937.7605      1.0000   669  17:08:00     0.8

Finished on: Fri Nov 30 17:08:00 2012

LogLikelihood Converged
> summary(kiwi.asr)$varcomp
                       gamma    component std.error   z.ratio constraint
Row!Row.var     3.236623e-01 3.236623e-01 0.2397829 1.3498137   Positive
Bay!Bay.var     3.933585e-02 3.933585e-02 0.1011562 0.3888624   Positive
Row:Bay!Row.var 4.496232e-01 4.496232e-01 0.2373959 1.8939807   Positive
ped(Vine)!ped   4.714762e-07 4.714762e-07        NA        NA   Boundary
ped(DVine)!ped  5.735897e-01 5.735897e-01 0.5565144 1.0306827   Positive
R!variance      1.000000e+00 1.000000e+00        NA        NA      Fixed
R!units.var     4.994783e+00 4.994783e+00 0.5330313 9.3705258   Positive
>
```

The results suggest that there is no additive genetic variance, though all of the other variance components are small relative to the residual error. Examination of various diagnostics failed to satisfactorily determine the cause of the relatively high residual variance. The code below presents one of the diagnostics we used in this examination. The *aom* argument of **ASReml** is used here. This (relatively) new (and mostly undocumented) facility will be discussed in detail at a talk we will present at this conference.

```
temp.asr <- update(kiwi.asr,aom=T)
temp.df <- kiwi.df
temp.df$et <- temp.asr$aom$R[,'stdCondRes']
subset(temp.df,abs(et)>3)
```

Figure 3.3 presents a plot of the studentised conditional residuals against vine number (within row) for each *Row*. Note that only female vines have data and so there are often large and variable gaps between the female vines within a row, although there are male vines in these gaps.

Rather than persist with this trait we consider fruit weight per fruit. Using the same model we get the following results:

```
> kiwi.asr1 <- asreml(FW11~1+Block,random=~Row + Bay + Row:Bay +
+                     ped(Vine) + ped(DVine),data=kiwi.df,na.method.X='include',
```

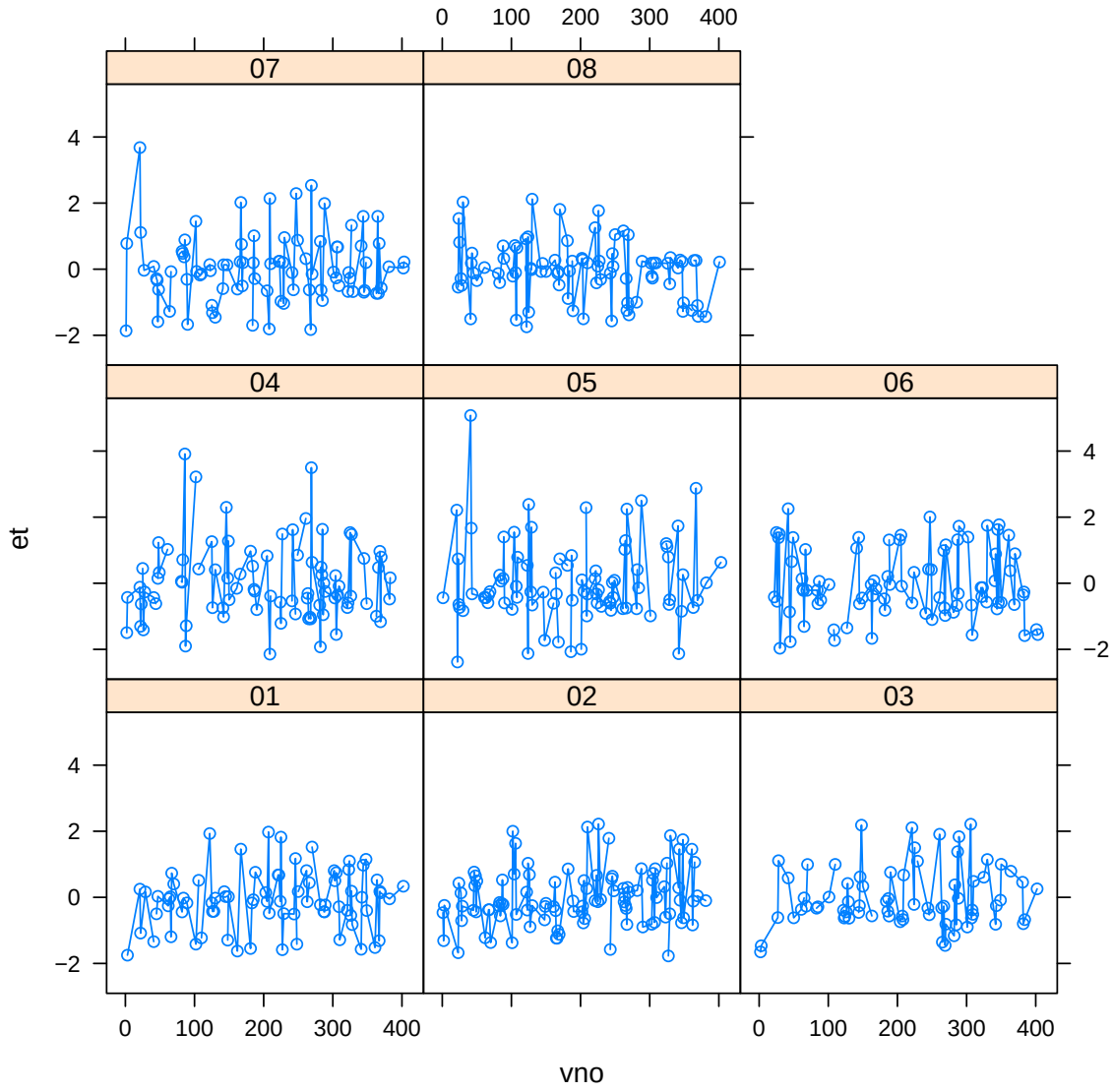

Figure 3.3: Plot of studentised conditional residuals against vine number for each orchard row of square root transformed fruit number per vine.

```
+                     rcov=~idv(units),ginverse=list(Vine=kiwi.ginv,DVine=kiwi.dinv),
+                      maxiter=15)

asreml 3.0-1 (24 October 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik          S2      DF      wall     cpu
  -6956.5659      1.0000   641  17:12:42     1.4
  -5652.6082      1.0000   641  17:12:43     0.9
  -4194.3769      1.0000   641  17:12:44     0.8
  -2960.8702      1.0000   641  17:12:44     0.8
  -2431.7940      1.0000   641  17:12:45     0.8
  -2235.6576      1.0000   641  17:12:46     0.9
  -2178.1430      1.0000   641  17:12:47     0.9
  -2163.0954      1.0000   641  17:12:48     0.8
  -2157.8011      1.0000   641  17:12:49     0.9
  -2155.9824      1.0000   641  17:12:49     0.8
  -2155.5323      1.0000   641  17:12:50     0.8
  -2155.4721      1.0000   641  17:12:51     0.8
  -2155.4692      1.0000   641  17:12:52     0.8
  -2155.4691      1.0000   641  17:12:53     1.0
  -2155.4691      1.0000   641  17:12:54     0.9

Finished on: Fri Nov 30 17:12:54 2012

LogLikelihood Converged
> summary(kiwi.asr1)$varcomp
                     gamma  component  std.error    z.ratio constraint
Row!Row.var       8.713664   8.713664   7.222884 1.20639684   Positive
Bay!Bay.var       0.118430   0.118430   3.431435 0.03451325   Positive
Row:Bay!Row.var   1.673516   1.673516  10.044233 0.16661458   Positive
ped(Vine)!ped   242.543971 242.543971 189.390070 1.28065833   Positive
ped(DVine)!ped   97.825086  97.825086 108.362251 0.90275982   Positive
R!variance        1.000000   1.000000         NA         NA      Fixed
R!units.var      93.483851  93.483851  95.573351 0.97813721   Positive
```

The results are far more encouraging with both relatively high additive and dominance genetic variance. The default plot using **plot.asreml** is presented in figure 3.4 and this suggests that there are several potentially *interesting* data values. These values are:

```
> temp.asr1 <- update(kiwi.asr1,aom=T,maxiter=1)

asreml 3.0-1 (24 October 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik          S2      DF      wall     cpu
  -2155.4691     93.4842   641  17:17:25     5.7

Finished on: Thu Nov 29 17:17:25 2012


Warning message:
In sqrt(asr$soln[, 3]) : NaNs produced
```

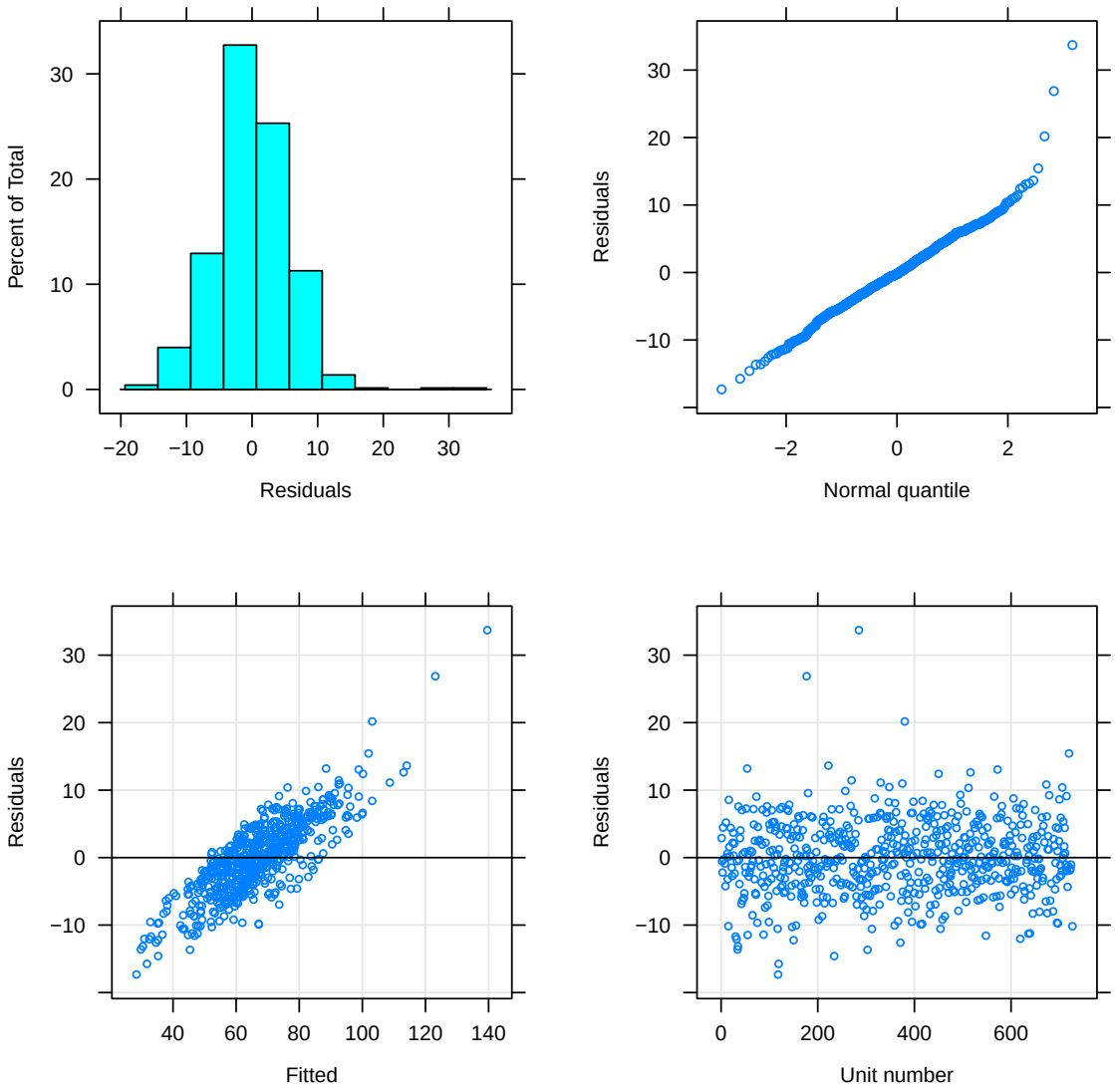

Figure 3.4: Default plot for the analysis of fruit weight per fruit.

```
> temp.df <- kiwi.df
> temp.df$et <- temp.asr1$aom$R[,'stdCondRes']
> subset(temp.df,abs(et)>3)[,c('Vine','ID','FN11','FW11','et')]
          Vine ID FN11  FW11        et
483 12E-02-06c A2   21  11.1 -3.191626
693 12E-02-16j B1   98 150.0  4.859490
694 12E-04-02i A2   48 173.3  6.205154
319 12E-05-04c A5   43 123.3  3.719839
```

In the absence of additional information, however do not discard these data.

We now present an alternate approach which does not incorporate pedigree information but uses the parental information alone. Our main purpose is more to illustrate the use of some model syntax facilities which are not commonly used or perhaps not well understood. The analyses we use produce an approximate analysis which may be useful as a starting point for the above more sophisticated modelling. There are similarities to the so-called reduced animal model which was used for variance components estimation in pig survival data using **ASReml** (White et al., 2006).

To undertake this analysis we need to create two factors, one for maternal the other for paternal coding, but each factor ***must*** have the identical levels. The following output presents how this may be achieved:

```
> all.names <- with(kiwi.df,c(paste('Mum',levels(factor(Mother)),sep=''),
+                             paste('Dad',levels(factor(Father)),sep='')))
> tmpm <- with(kiwi.df,paste('Mum',Mother,sep=''))
> tmpf <- with(kiwi.df,paste('Dad',Father,sep=''))
> table(tmpm,tmpf)
      tmpf
tmpm   Dad1 Dad2 Dad3 Dad4 Dad5 Dad6 Dad7 Dad8
  Mum1   40   50   42   47   42   48    0    0
  Mum2   45   42   51   49   38   49    0    0
  Mum3    0    0    0    0    0    0   46   51
  Mum4    0    0    0    0    0    0   44   43
> kiwi.df$Mum <- factor(tmpm,levels=all.names)
> kiwi.df$Dad <- factor(tmpf,levels=all.names)
> with(kiwi.df,table(Mum,Dad))
      Dad
Mum    Mum1 Mum2 Mum3 Mum4 Dad1 Dad2 Dad3 Dad4 Dad5 Dad6 Dad7 Dad8
  Mum1    0    0    0    0   40   50   42   47   42   48    0    0
  Mum2    0    0    0    0   45   42   51   49   38   49    0    0
  Mum3    0    0    0    0    0    0    0    0    0    0   46   51
  Mum4    0    0    0    0    0    0    0    0    0    0   44   43
  Dad1    0    0    0    0    0    0    0    0    0    0    0    0
  Dad2    0    0    0    0    0    0    0    0    0    0    0    0
  Dad3    0    0    0    0    0    0    0    0    0    0    0    0
  Dad4    0    0    0    0    0    0    0    0    0    0    0    0
  Dad5    0    0    0    0    0    0    0    0    0    0    0    0
```

```
Dad6    0    0    0    0    0    0    0    0    0    0    0    0
Dad7    0    0    0    0    0    0    0    0    0    0    0    0
Dad8    0    0    0    0    0    0    0    0    0    0    0    0
```

Note that each maternal and paternal factors have identical levels.

The syntax for this is:

```
kiwi.asr2 <- asreml(FW11~1+Block,random=~Row + Bay + Row:Bay +
                    zero:Mum + and(Mum,0.5) + and(Dad,0.5) + ID,
                    data=kiwi.df,na.method.X='include',rcov=~units)
```

where we include *Mum* and *Dad* using the *and()* syntax which overlays the two design matrices. Note that an additional *dummy* term has to be included before the first *and()* to allow the next term to be overlaid onto it.

We now present an alternate but perhaps interesting syntax which has been recently developed for specifying variance complex models, including random regression models. The special variance function is *str* and is described in detail in the **ASReml-R** manual in section 4.3.6. The two arguments specify a model formula (not containing terms with any variance model) and a model formula containing the variance model to apply to the model formula.

```
kiwi.asr3 <- asreml(FW11~1+Block,random=~Row + Bay + Row:Bay +
                    str(~zero:Mum + and(Mum,0.5) + and(Dad,0.5),~idv(Mum)) + ID,
                    data=kiwi.df,na.method.X='include',rcov=~idv(units))
```

In this simplified model we have excluded the dominance effects and have included *ID*. This term can be thought of as a surrogate for the dominance effects. There is again an additional term which hs ignored in the approximate analysis which can be shown to be have approximate variance of $3\sigma_d^2/4$. This variance along with the Mendelian sampling variance which has been implicity ignored by use of the parental factors in place of the pedigree information for individuals, become part of the residual variance and hence an estimate of the residual variance from the approximate analysis can be simply computed which we illustrate in the following. The results from the approximate analysis are:

```
> kiwi.asr2 <- asreml(FW11~1+Block,random=~Row + Bay + Row:Bay +
+                     zero:Mum + and(Mum,0.5) + and(Dad,0.5) + ID,
+                     data=kiwi.df,na.method.X='include',rcov=~idv(units))

asreml 3.0-1 (24 October 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik          S2      DF      wall     cpu
  -6863.8844      1.0000   641  17:32:55     0.0
  -5581.9381      1.0000   641  17:32:55     0.0
  -4148.7497      1.0000   641  17:32:55     0.0
  -2937.4835      1.0000   641  17:32:55     0.0
  -2419.5098      1.0000   641  17:32:55     0.0
  -2229.0524      1.0000   641  17:32:55     0.0 (1 restrained)
```

```
  -2181.9564      1.0000   641  17:32:55     0.0
  -2163.7227      1.0000   641  17:32:55     0.0
  -2158.7085      1.0000   641  17:32:55     0.0
  -2156.9737      1.0000   641  17:32:55     0.0
  -2156.5878      1.0000   641  17:32:55     0.0
  -2156.5524      1.0000   641  17:32:55     0.0
  -2156.5519      1.0000   641  17:32:55     0.0

Finished on: Fri Nov 30 17:32:55 2012

LogLikelihood not converged
> summary(kiwi.asr2)$varcomp
                         gamma    component  std.error     z.ratio constraint
Row!Row.var          8.71737555   8.71737555   7.257128  1.20121561   Positive
Bay!Bay.var          0.07615419   0.07615419   3.453866  0.02204897   Positive
Row:Bay!Row.var      2.27422933   2.27422933  10.161114  0.22381693   Positive
zero:Mum!zero.var 195.03185754 195.03185754 112.542722  1.73295842   Positive
ID!ID.var            7.44514370   7.44514370   8.742678  0.85158614   Positive
R!variance           1.00000000   1.00000000         NA          NA      Fixed
R!units.var        284.25900815 284.25900815  18.005451 15.78738649   Positive
```

Notice the reduction in time per iteration, even for this relatively small data-set and pedigree. The estimated variance components are similar, though as expected there is a consistent negative bias for the approximate analysis.

3.3.5 Summarising the analyses

To conclude the analysis of this example we now illustrate how to use the **predict** method for **ASReml** objects to produce estimated breeding values and accuracies. The same information can be extracted using the *summary* method for **ASReml** but it can be tedious, clumsy and error prone.

```
> tt <- summary(kiwi.asr1,all=T)
> head(tt$coef.random[grep('ped\\(Vine\\.*)',dimnames(tt$coef.random)[[1]]),],25)
                            solution std error       z ratio
ped(Vine)_CK01           1.464045e+00  15.34766  9.539208e-02
ped(Vine)_CK15          -3.034342e-01  15.19639 -1.996752e-02
ped(Vine)_CK51          -1.713092e-10  15.57304 -1.100037e-11
ped(Vine)_CK65          -6.463049e+00  14.80913 -4.364232e-01
ped(Vine)_CK64           7.069917e+00  14.65916  4.822867e-01
ped(Vine)_CK65_09       -1.292610e+01  12.23448 -1.056531e+00
ped(Vine)_CK64_18        1.413983e+01  11.48933  1.230692e+00
ped(Vine)_CK15_03       -5.454155e+00  14.64788 -3.723510e-01
ped(Vine)_CK15_01        4.695569e+00  14.78863  3.175120e-01
ped(Vine)_CK51_06       -6.859193e+00  14.41111 -4.759657e-01
ped(Vine)_CK51_09        3.231776e+00  12.89658  2.505917e-01
ped(Vine)_CK51_11        3.627417e+00  12.62104  2.874102e-01
ped(Vine)_CK01_01_01_01  2.928090e+00  14.65071  1.998599e-01
```

```
ped(Vine)_CK01_02_12_02 -3.838393e+00  14.53716 -2.640401e-01
ped(Vine)_52-13-22d      7.529702e+00  14.50473  5.191206e-01
ped(Vine)_Hort16A        1.027488e+01  12.21301  8.413062e-01
ped(Vine)_30-3-10f      -2.586068e+00  11.63801 -2.222087e-01
ped(Vine)_30-4-8b        2.491258e+00  11.60910  2.145953e-01
ped(Vine)_30-4-8c       -2.956791e+00  11.60855 -2.547080e-01
ped(Vine)_30-4-8d       -1.625131e+01  11.61793 -1.398813e+00
ped(Vine)_30-5-10b      -1.099838e+01  11.64589 -9.443997e-01
ped(Vine)_A1            -4.722630e+00  14.38745 -3.282465e-01
ped(Vine)_A2            -2.183967e+00  14.38092 -1.518657e-01
ped(Vine)_A3            -4.907992e+00  14.38338 -3.412266e-01
ped(Vine)_A4            -1.155525e+01  14.38415 -8.033320e-01
> term.use <- kiwi.asr1$factor.names[6]
> male.names <- levels(factor(kiwiped.df$Male))
> male.names <- male.names[table(kiwiped.df$Male)>5]
> kiwi.pvs <- predict(kiwi.asr1,classify='ped(Vine)',only=term.use,
+                     maxiter=1,levels=list('ped(Vine)'=male.names),
+                     vcov=T)

asreml 3.0-1 (24 October 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik          S2      DF      wall     cpu
  -2155.4691     93.4842   641  23:58:52     1.4

Finished on: Thu Nov 29 23:58:52 2012


> male.ebvs <- kiwi.pvs$predictions$pvals
> siga <- summary(kiwi.asr1)$varcomp['ped(Vine)!ped','component']
> male.ebvs$acc <- sqrt(1 - male.ebvs$standard.error^2/siga)
> male.ebvs

Notes:
- The predictions are obtained by averaging across the hypertable
  calculated from model terms constructed solely from factors in
  the averaging and classify sets.
- Use "average" to move ignored factors into the averaging set.

- The SIMPLE averaging set:  Block mv
- The ignored set:  Row Bay ped(DVine)
- ped(Vine) is included in this prediction

      Vine predicted.value standard.error est.status       acc
1 30-3-10f       -2.586154       11.63845  Estimable 0.6645302
2  30-4-8b        2.491341       11.60953  Estimable 0.6666121
3  30-4-8c       -2.956869       11.60898  Estimable 0.6666516
4  30-4-8d      -16.251799       11.61836  Estimable 0.6659778
5 30-5-10b      -10.998755       11.64633  Estimable 0.6639610
6  CK51_09        3.231907       12.89714  Estimable 0.5606143
7  CK51_11        3.627447       12.62160  Estimable 0.5858963
8  CK64_18       14.140245       11.48977  Estimable 0.6751124
```

4 Mixed models for Geostatistics

4.1 Introduction

Geostatistics has evolved as a branch of spatial statistics concerned primarily with the prediction of a spatially dependent quantity based on observations at a set of (pre-specified) locations.

Geostatistics as a subject in its own right developed from the initial work of Georges Matheron and colleagues at Fontainebleau, France. These ideas were developed independently of other work in spatial statistics, and the division between geostatistics and main stream spatial statistics is illustrated by the basic geostatistical tool known as kriging. It is now well known that kriging (after Krige (1951)) is equivalent to minimum mean square error prediction under a (Gaussian) linear mixed model. This connection has been made on various occasions since Ripley (1981) and was a primary motivation for the excellent theoretical treatment of the subject in Smith (1999). Recently Diggle et al. (1998) coined the phrase *model-based geostatistics* to refer to an approach to geostatistical problems that uses the application of formal statistical methods based on an assumed stochastic model. We adopt this approach in the following but narrow our focus to the Gaussian setting.

4.2 Motivating example: electromagnetic salinity

Rapid and cost-effective measurement of soil salinity via apparent electrical conductivity (EC) of soil profiles is becoming an important land management tool. The current protocol involves measurements of EC from a ground-based electromagnetic induction instrument, which is linked to a differential global positioning system and towed behind a four-wheel motorbike, providing a large number of geographically-referenced observations. The data set used here was kindly provided by Jo Stringer and was collected by Independent Agricultural Resources Pty Ltd based in Mackay in central Qld, who work with BSES in the Australian Sugar Industry. From one field, 2780 observations were gathered in a serpentine fashion, as displayed in Figure 4.1. All of the locations were distinct. The aim of the current analysis is to understand the characteristics of the spatial variation.

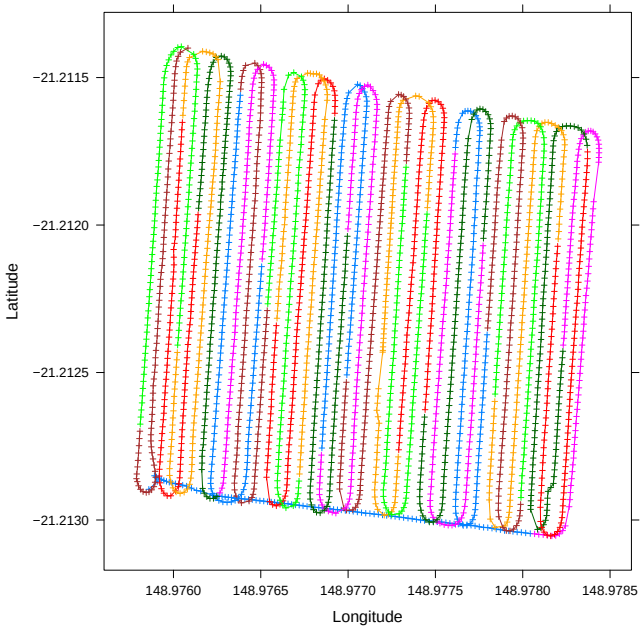

Figure 4.1: Layout of EC measurements within field. Colour changes every 100 measurements to show order of measurement. Start is at blue points running along bottom of plot.

4.3 Geostatistical mixed model

We assume that we have observed data at a set of n locations (of which b, possibly less than n, are distinct), with the ith observation y_i taken at the location identified by a vector $\boldsymbol{s}_i$, $i = 1 \ldots n$.

A model for y_i is

$$y_i = f(\boldsymbol{s}_i) + e_i \tag{4.3.1}$$

where $f(\boldsymbol{s}_i)$ is some function of the spatial location $\boldsymbol{s}_i$ and the e_i are mutually independent N(0, σ^2) random variables. If $\boldsymbol{s}$ represents the set of b distinct observed locations, and $\boldsymbol{f}(\boldsymbol{s}) = (f(\boldsymbol{s}_1), \ldots, f(\boldsymbol{s}_n))'$, then we assume

$$\boldsymbol{f}(\boldsymbol{s}) = \boldsymbol{X}_s\boldsymbol{\tau}_s + \boldsymbol{Z}_s\boldsymbol{u}_s(\boldsymbol{s}) \tag{4.3.2}$$

where $\boldsymbol{X}$ is an $n \times p$ matrix of polynomials in $\boldsymbol{s}$, often of degree 1 and associated vector $\boldsymbol{\tau}_s$ is a $p \times 1$ vector of polynomial regression coefficients. This term is included in the model to account for so-called *global trend* or non-stationary behaviour. The matrix $\boldsymbol{Z}$ is an $n \times b$ indicator matrix for random effects at distinct locations, which can accommodate duplicated locations (although typically $\boldsymbol{Z}_s = \boldsymbol{I}_n$ if all the locations are distinct). Finally $\boldsymbol{u}_s(\boldsymbol{s})$ is a realisation of a stationary Gaussian process distributed independently of $\boldsymbol{e} = (e_1, \ldots, e_n)'$, with zero mean and variance matrix $\sigma_s^2\boldsymbol{G}_s(\boldsymbol{\phi}_s)$. The matrix $\boldsymbol{G}_s(\boldsymbol{\phi}_s)$ is a positive definite correlation matrix with elements given by $\rho(\boldsymbol{s}_i - \boldsymbol{s}_j, \boldsymbol{\phi}_s)$, $\rho(\cdot)$ being a correlation function with a parameter vector $\boldsymbol{\phi}_s$ and dependent on the spatial separation vector $\boldsymbol{h}_{ij} = \boldsymbol{s}_i - \boldsymbol{s}_j$. Subscripts s indicate elements specifically related to spatial effects.

Combining equations (4.3.1) and (4.3.2) we have, in matrix notation,

$$\boldsymbol{y}(\boldsymbol{s}) = \boldsymbol{X}_s\boldsymbol{\tau}_s + \boldsymbol{Z}_s\boldsymbol{u}_s(\boldsymbol{s}) + \boldsymbol{e} \tag{4.3.3}$$

and, with estimation on the component scale,

$$\boldsymbol{y}(\boldsymbol{s}) \sim \mathrm{N}(\boldsymbol{X}_s\boldsymbol{\tau}_s, \sigma_s^2\boldsymbol{Z}_s\boldsymbol{G}_s\boldsymbol{Z}_s' + \sigma_e^2\boldsymbol{I}_n)\,. \tag{4.3.4}$$

In the notation of chapter 1, we have

$$\boldsymbol{\sigma}_{gv} = (\sigma_s^2), \quad \boldsymbol{\sigma}_{gc} = \boldsymbol{\phi}_s$$
$$\boldsymbol{\sigma}_{rv} = (\sigma_e^2)$$

and $\boldsymbol{\sigma}_{rc}$ is a null vector. With estimation on the ratio scale, we have

$$\boldsymbol{y}(\boldsymbol{s}) \sim \mathrm{N}(\boldsymbol{X}_s\boldsymbol{\tau}_s, \theta(\gamma_s\boldsymbol{Z}_s\boldsymbol{G}_s\boldsymbol{Z}_s' + \boldsymbol{I}_n)) \tag{4.3.5}$$

where $\theta = \sigma_e^2$ and $\gamma_s = \sigma_s^2/\sigma_e^2$. We then have

$$\boldsymbol{\gamma}_{gv} = (\gamma_s), \quad \boldsymbol{\gamma}_{gc} = (\boldsymbol{\phi}_s)$$

and both $\boldsymbol{\gamma}_{rv}$ and $\boldsymbol{\gamma}_{rc}$ are null vectors. In geostatistical terminology, the IID error term in (4.3.4) and (4.3.5) models a nugget effect.

In the case that the n locations are distinct ($b = n$), then $\boldsymbol{Z}_s = \boldsymbol{I}_n$ in the equations above, and the roles of $\boldsymbol{e}$ and $\boldsymbol{u}$ can be switched. Note that in many ways this distinction is artificial and made only to comply with the conventions of mixed model packages. The model can then be formulated with spatially correlated residual errors $\boldsymbol{e}_s(\boldsymbol{s}) \sim N(\boldsymbol{0}, \sigma_s^2\boldsymbol{R}_s(\boldsymbol{\phi}_s))$, with the nugget effect modelled as an independent random $n \times 1$ effect $\boldsymbol{u} \sim N(\boldsymbol{0}, \sigma_e^2\boldsymbol{I}_n)$, with design matrix $\boldsymbol{I}_n$, so

$$\boldsymbol{y}(\boldsymbol{s}) = \boldsymbol{X}_s\boldsymbol{\tau}_s + \boldsymbol{u} + \boldsymbol{e}_s \tag{4.3.6}$$

and, with estimation on the component scale,

$$\boldsymbol{y}(\boldsymbol{s}) \sim \mathrm{N}(\boldsymbol{X}_s\boldsymbol{\tau}_s, \sigma_e^2\boldsymbol{I}_n + \sigma_s^2\boldsymbol{R}_s) \tag{4.3.7}$$

where $\boldsymbol{R}_s = \boldsymbol{G}_s$, is now renamed to reflect the allocation to the R-structure. Now in our standard notation we have

$$\boldsymbol{\sigma}_{gv} = (\sigma_e^2),$$
$$\boldsymbol{\sigma}_{rv} = (\sigma_s^2), \quad \boldsymbol{\sigma}_{rc} = \boldsymbol{\phi}_s$$

and $\boldsymbol{\sigma}_{gc}$ is a null vector. Switching back again to estimation on the ratio scale, we have

$$\boldsymbol{y}(\boldsymbol{s}) \sim \mathrm{N}(\boldsymbol{X}_s\boldsymbol{\tau}_s, \theta_s(\gamma_e\boldsymbol{I}_n + \boldsymbol{R}_s)) \tag{4.3.8}$$

where $\theta_s = \sigma_s^2$ and $\gamma_e = \sigma_e^2/\sigma_s^2$. We use the terminology θ_s here to stress that the spatial structure is now the residual term and to distinguish it from θ in Equation 4.3.5, which was related to the nugget term. We then have

$$\boldsymbol{\gamma}_{gv} = (\gamma_e)$$
$$\boldsymbol{\gamma}_{rc} = \boldsymbol{\phi}_s$$

and both $\boldsymbol{\gamma}_{gc}$ and $\boldsymbol{\gamma}_{rv}$ are null vectors.

We have now constructed four forms of this model, which appear different but are in fact equivalent. This ambiguity of form can lead to some confusion, and so is important to recognise. One of the first two forms (4.3.4 or 4.3.5) must be used when there are locations with multiple observations, and the one of the second two forms (4.3.7 or 4.3.8) must be used to fit a model with no nugget effect. Recall that it is not possible to use the *asreml* function to fit a model without a nugget effect for data with multiple observations at a single location.

We will interchange between these two forms in the analysis of our example and it is therefore important to understand the equivalence between them.

4.4 Covariance models for Gaussian random fields

The mathematical description of the dependence between observations at different locations is central to geostatistics and hence the key element of our geostatistical model is the specification of the covariance model for $\boldsymbol{u}_s(\boldsymbol{s})$. In this section we will briefly review some aspects of the theory of random fields, common nomenclature and results. A more thorough account can be found in Smith (1999).

4.4.1 Preliminaries

In the following for ease of notation we shall drop the subscript and denote the random field $U_s(\boldsymbol{s})$ by $U(\boldsymbol{s})$, which is Gaussian if the joint distribution of $U(\boldsymbol{s}_1), \ldots, U(\boldsymbol{s}_n)$ is multivariate normal for any n set of locations. We denote the covariance function by $\psi(\cdot, \cdot)$ and this function must satisfy

$$\sum_{i,j=1}^{n} c_i c_j \psi(\boldsymbol{s}_i, \boldsymbol{s}_j) \geq 0$$

in order to give a positive definite variance function. We denote $m(\boldsymbol{s})$ as the mean of $U(\boldsymbol{s})$, then the joint distribution of $U(\boldsymbol{s}_1), \ldots, U(\boldsymbol{s}_n)$ is multivariate normal with mean $(m(\boldsymbol{s}_1), \ldots, m(\boldsymbol{s}_n))'$ and covariance matrix Ψ with ijth element given by $\psi(\boldsymbol{s}_i, \boldsymbol{s}_j)$.

4.4.2 Stationarity

Since observations are made from only a single realisation of a random field we cannot make much progress without further assumptions. Stationarity is an important simplifying assumption. There are several forms of stationarity. A random field is said to be weakly (or second order) stationary if

- its mean is independent of its location, ie. $m(\boldsymbol{s}) = \mu$ say, for all $\boldsymbol{s} \in \boldsymbol{R}^2$
- the covariance function between any pair of locations $\boldsymbol{s}$ and $\boldsymbol{t}$ is a function only of the the spatial separation vector $\boldsymbol{h} = \boldsymbol{s} - \boldsymbol{t}$. That is

$$\psi(\boldsymbol{s}, \boldsymbol{t}) = \psi(\boldsymbol{h})$$

Using this assumption we can characterize the random field by its correlation function $\rho(\cdot)$ with $\psi(\boldsymbol{h}) = \sigma_s^2 \rho(\boldsymbol{h})$, noting that $\rho(0) = 1$.

4.4.3 Isotropy

Another important property of a random field is isotropy. A weakly stationary random field (in more than one dimension) is said to be isotropic if the dependence between any pair of observations depends only on the Euclidean distance between them. Otherwise it is said to be anisotropic. In terms of the correlation function, then

$$\rho(\boldsymbol{s}, \boldsymbol{t}) = \rho(\boldsymbol{h}) = \rho(\boldsymbol{d})$$

where $\boldsymbol{d} = ||\boldsymbol{h}||$.

4.4.4 The variogram

Traditional geostatistics relies heavily on the *variogram*. The variogram exists and is linked to the covariance function for weakly stationary random fields. The variogram of a random field $U(\boldsymbol{s})$ is defined to be the function $V(\boldsymbol{s}, \boldsymbol{t}) = \frac{1}{2}\text{var}\left(U(\boldsymbol{s}) - U(\boldsymbol{t})\right)$ for any $\boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{R}^2$. If the random field is weakly stationary then this reduces to

$$\begin{aligned} V(\boldsymbol{s}, \boldsymbol{t}) = V(\boldsymbol{h}) &= \frac{1}{2}\text{var}\left(U(\boldsymbol{s}) - U(\boldsymbol{t})\right) \\ &= \frac{1}{2}\text{E}\left(\{U(\boldsymbol{s}) - U(\boldsymbol{t})\}^2\right) \\ &= \sigma_s^2(1 - \rho(\boldsymbol{h})) \end{aligned}$$

If the random field is also isotropic then the variogram is a function only of d the Euclidean distance.

Technically $2V(\boldsymbol{h})$ is called the variogram and $V(\boldsymbol{h})$ the semi-variogram, but more often the term variogram has been used for $V(\cdot)$ and we will use this convention.

4.4.5 Geometric Anisotropy

When considering observations taken at spatial locations in two (or higher) dimensional space, we may wish to retain the assumption of weak stationarity but avoid the assumption of isotropy. This amounts to relaxing the assumption that the covariance function $\psi(\cdot)$ is a function only of the Euclidean distance. In $\boldsymbol{R}^2$ isotropic correlation (or covariance) has circular contours of constant correlation with respect to the elements of the spatial separation vector $\boldsymbol{h} = (h_1, h_2)'$. Geometric anisotropy allows ellipsoid contours of constant correlation in two dimensions, which need not be aligned with the coordinate axes. Geometric anisotropy in two dimensions can be specified via a transformation of $\boldsymbol{h}$ which depends on an anisotropy angle α and an anisotropy ratio δ. Higher dimensional geometric anisotropy requires more parameters to be completely general.

The correlation function of an isotropic random field is a function only of the Euclidean distance d. To convert this correlation function $\rho(\cdot)$ to geometric anisotropy we apply a rotation of the

original coordinates through α radians then stretch (or shrink) the resulting axes relative to each other. In matrix notation we have

$$\boldsymbol{h}^{**} = \begin{bmatrix} h_1^{**} \\ h_2^{**} \end{bmatrix} = \begin{bmatrix} \sqrt{\delta} & 0 \\ 0 & \frac{1}{\sqrt{\delta}} \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \boldsymbol{S}\boldsymbol{h}^{*} = \boldsymbol{S}\boldsymbol{T}\boldsymbol{h}$$

where $\boldsymbol{h}^{*} = \boldsymbol{T}\boldsymbol{h}$. The geometric anisotropic correlation function is then a function of the Euclidean distance based on $\boldsymbol{h}^{**}$, that is

$$d^2 = \boldsymbol{h}^{**\prime}\boldsymbol{h}^{**} = \boldsymbol{h}^{\prime}\boldsymbol{T}^{\prime}\boldsymbol{S}^2\boldsymbol{T}\boldsymbol{h}$$

We note that there is non-uniqueness in this metric $d(\cdot)$, since inverting δ and adding $\frac{\pi}{2}$ to α gives the same distance. This non-uniqueness can be removed by constraining $0 \leq \alpha < \frac{\pi}{2}$ and $\delta > 0$, or by constraining $0 \leq \alpha < \pi$ and either $0 < \delta \leq 1$ or $\delta \geq 1$. Isotropy corresponds to $\delta = 1$, and then the rotation angle α is irrelevant.

4.4.6 Minkowski metric

The anisotropic correlation function described in Section 4.4.5 can be further generalised (for random fields in more than one dimension) by replacing the usual Euclidean metric by the so-called Minkowski metric. The Minkowski metric applied to the transformed coordinates is

$$d(\boldsymbol{h}; \delta, \alpha, \lambda) = \left(\delta |h_1^{*}|^{\lambda} + \frac{1}{\delta} |h_2^{*}|^{\lambda} \right)^{1/\lambda}$$

recalling

$$\boldsymbol{h}^{*} = \begin{bmatrix} h_1^{*} \\ h_2^{*} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \boldsymbol{T}\boldsymbol{h}$$

and with λ usually taken to be a positive integer. When $\lambda = 2$ this metric is the Euclidean metric and when $\lambda = 1$ it corresponds to the city-block metric used in the analysis of field trials (Cullis & Gleeson, 1991). Following Haskard et al. (2007) we can then embed this generalised metric into the correlation function $\rho(\cdot)$ giving $\rho(\boldsymbol{h}) = \rho(d(\boldsymbol{h}; \delta, \alpha, \lambda))$.

4.4.7 Parametric correlation models

Many parametric correlation models have been suggested for use in geostatistics. Some of those available in *asreml* include:

exponential model:

$$\rho(d) = \exp(-d/\phi)$$

gaussian model:

$$\rho(d) = \exp(-(d/\phi)^2)$$

spherical model:

$$\rho(d) = \begin{cases} 1 - \frac{3}{2}(d/\phi) + \frac{1}{2}(d/\phi)^3 & \text{if } 0 \leq d < \phi, \\ 0 & \text{if } d \geq \phi \end{cases}$$

circular model:

$$\rho(d) = \begin{cases} \frac{2}{\pi}\cos^{-1}(d/\phi) - \frac{d}{\phi}\sqrt{1-(\frac{d}{\phi})^2} & \text{if } 0 \le d < \phi, \\ 0 & \text{if } d \ge \phi \end{cases}$$

Following the recommendations of Smith (1999) we prefer to use the Matérn family of correlation functions, which are implemented in *asreml* in the generality of Section 4.4.6. The basic isotropic Matérn correlation function is given by

$$\rho_M(d;\phi,\nu) = \left\{2^{\nu-1}\Gamma(\nu)\right\}^{-1}\left(\frac{d}{\phi}\right)^{\nu} K_\nu\left(\frac{d}{\phi}\right), \tag{4.4.9}$$

where $\phi > 0$ is a range parameter, $\nu > 0$ is a smoothness parameter, $\Gamma(\cdot)$ is the gamma function, and $K_\nu(.)$ is the modified Bessel function of the third kind of order ν (Abramowitz & Stegun, 1965, $S9.6$). For a given ν, the range parameter ϕ affects the rate of decay of $\rho(\cdot)$ with increasing d. The parameter $\nu > 0$ controls the analytic smoothness of the underlying process $\boldsymbol{u}_s$, the process being $\lceil\nu\rceil - 1$ times mean-square differentiable, where $\lceil\nu\rceil$ is the smallest integer greater than or equal to ν (Smith, 1999, p. 31). Larger ν correspond to smoother processes.

When $\nu = m + \frac{1}{2}$ with m a non-negative integer $\rho_M(\cdot)$ is then the product of $\exp(-d/\phi)$ and a polynomial of degree m in d. Thus if $\nu = \frac{1}{2}$ then we get the the exponential correlation function, $\rho_M(d;\phi,\frac{1}{2}) = \exp(-d/\phi)$, while $\nu = 1$ yields Whittle's elementary correlation function, $\rho_M(d;\phi,1) = (d/\phi)K_1(d/\phi)$ (Webster & Oliver, 2001, p. 119). When $\nu = 1.5$ then

$$\rho_M(d;\phi,1.5) = \exp(-d/\phi)(1 + d/\phi)$$

which is the correlation function of a random field which is continuous and once differentiable. This has been used recently by Kammann & Wand (2003). As $\nu \to \infty$ then $\rho_M(\cdot)$ tends to the gaussian correlation function.

Thus the Matérn correlation function offers flexibility and parsimony and includes many other correlation functions as special cases.

4.4.8 Extended geometric anisotropy within the Matérn class

We will use the correlation model suggested by Haskard (2005) (and implemented in the *asreml* function) which is the Matérn family of correlation functions, incorporating geometric anisotropy and a choice of distance metrics. This is given by

$$\rho(\boldsymbol{h};\boldsymbol{\phi}) = \rho_M(d(\boldsymbol{h};\delta,\alpha,\lambda);\phi,\nu)$$

where $\boldsymbol{h} = (h_1,h_2)^T$ is the spatial separation vector, (δ,α,λ) govern the choice of metric and geometric anisotropy and (ϕ,ν) are the parameters of the Matérn correlation function. The metric parameter λ is usually set to either 1 or 2, primarily based on context, but mindful of the criticisms aimed at the use of separable correlation models for spatial data. Geometric anisotropy is discussed in most geostatistical books (Webster & Oliver, 2001; Diggle et al., 2003) but rarely are the anisotropy angle or ratio estimated from the data. Similarly the smoothness parameter

ν is often set a-priori (Kammann & Wand, 2003; Diggle et al., 2003), and we will follow this convention. Haskard et al. (2007) present a more thorough investigation of the properties of REML estimates of the parameters in the extended correlation model described above.

4.5 Model building and diagnostics

The other major advantage of the geostatistical linear mixed model (over classical geostatistics) is that the relative fit of different variance models can be assessed within the framework of REML likelihood ratio tests, as outlined in Chapter 2.

A test for (geometric) anisotropy deserves closer attention and was considered in some detail by Haskard (2005).

The formal model selection process is aided by use of various graphical tools based on either the original data or more often BLUPs or residuals from intermediate and final models, including graphical displays of the sample omni-directional or directional semi-variogram which are defined below.

4.5.1 Sample semi-variograms

The sample omni-directional semi-variogram is based on the empirical omni-directional semi-variogram of the BLUP of a random field $\tilde{\boldsymbol{u}}_s = (\tilde{\boldsymbol{u}}_s(\boldsymbol{s}_1), \ldots, \tilde{u}_s(\boldsymbol{s}_n))^T$ which is given by the set of points $(d_{ij}, \tilde{v}_{ij}) : j < i$ where $d_{ij} = ||\boldsymbol{s}_i - \boldsymbol{s}_j||$ and $\tilde{v}_{ij} = \frac{1}{2}(\tilde{u}_s(\boldsymbol{s}_i) - \tilde{u}_s(\boldsymbol{s}_j))^2$.

In a regular spatial sampling design, say an $r \times c$ array with equal (assumed to be r_1 and r_2) spacings, then the separation vectors and the set of pairwise distances take a smaller number of unique values. This suggests that it may be sensible to average the $\tilde{v}_{ij}$ for each of these distinct values of d_{ij}. For irregular spatial sampling designs there may be no replication of the unique values of the d_{ij} but the standard approach is to 'bin' the semi-variances according to the following principle. The sample omni-directional semi-variogram is the set of points $d_k, \bar{v}_k$ where the $d_k, k = 1, \ldots, q$ are pre-specified distances and

$$\bar{v}_k = \frac{1}{n_k} \sum_{d_{ij} \in S_k} \bar{v}_{ij}$$

where S_k is the set of points for which d is closer to d_k than any other $d_{k'}$ and n_k is the number of elements in S_k.

To examine anisotropy we need to consider graphical displays of the semi-variance in terms of the elements (or functions) of the spatial separation vector $\boldsymbol{h}$. The empirical semi-variogram cloud is defined to be the set of triples $(h_{ij1}, h_{ij2}, \tilde{v}_{ij})$. This is then graphically represented by two forms of 'binning' based either on the cartesian coordinates of (h_1, h_2) or the polar coordinates (d, t) where $d = \sqrt{h_1^2 + h_2^2}$ and $t = tan^{-1}(h_2/h_1)$. The former display based on cartesian co-ordinates has been widely used for the analysis of field trials. The latter graphical display, based on polar co-ordinates is usually more suitable for other applications of the geostatistical linear mixed model.

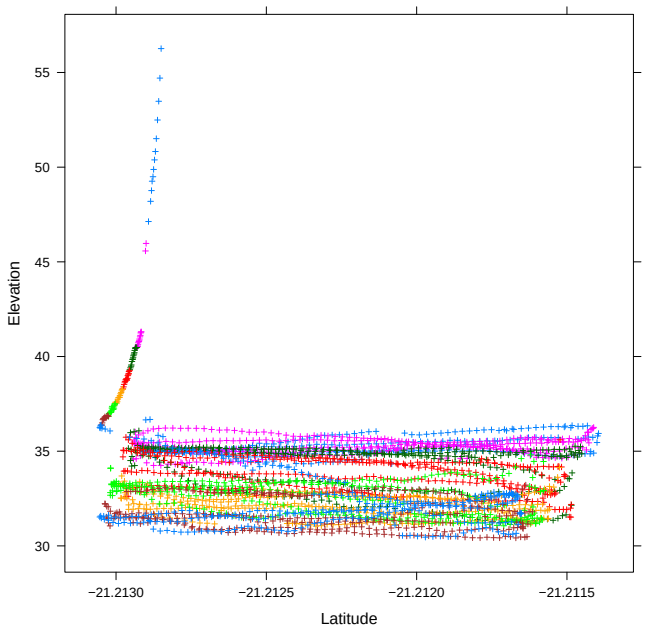

Figure 4.2: Elevation at measurement locations plotted against latitude, coloured by longitude groups.

4.6 Analysis of example: electromagnetic salinity

The data for this example is help in file EC.dat, which contains columns specifying the unit number (*unit*, giving the order of measurement), longitude (*long*) and latitude (*lat*) of measurement position, two readings at shallow (c) and deeper (d) levels and the elevation (*elevation*) for each measurement position. We follow the convention required by *asreml.read.table* of using lower case names for variates and so can read the data into dataframe *EC* using the command

```
EC <- asreml.read.table("EC.dat",header=T)
```

Here we analyse the shallower readings (c), with the aim of investigating spatial variation. Figure 4.1 showed the spatial location of the measurements. For models based on spatial co-ordinates, it can be helpful (in terms of scaling distance-dependent correlation parameters) to re-scale the co-ordinates, and so we do this here as

```
EC$along <- 100000*(EC$long-148.97)
EC$alat <- 100000*(EC$lat+21.21)
```

which gives median distances between successive points of 2.3 units of rescaled latitude and 0.3 units of rescaled longitude. We use the same rescaling factor in both dimensions to avoid spatial distortion.

We start by investigating the data and first consider the covariates, ie. spatial location and elevation. Figure 4.2 plots elevation against latitude, coloured by longitude groups (ie. longitude

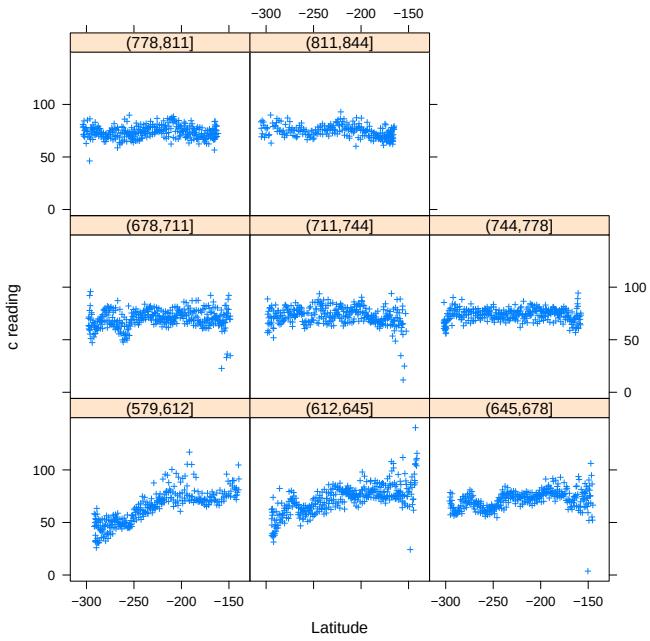

Figure 4.3: Reading c plotted against latitude, separated into longitude groups.

grouped into 8 sets). There are some unusually high values at the start of the series, and these are incongruent with the elevation measured at these locations later in the series. There is a (spatial) gap in the series after 156 measurements, and the elevation values appear less alarming after this point. This may indicate that the equipment was not properly initialised and so (in the absence of further information) we distrust and discard these data values. In practice of course we would investigate further before taking this step. This (and other) plots also indicate that elevation is reasonably constant across (most) latitudes, but appears to decrease as longitude increases, ie. the field is sloped moving downwards on the right side of the field.

We do the subsetting using the *subset* function (as below) and now work with the smaller data frame called *ec.df*.

```
ec.df <- subset(EC,unit>156
```

We now start looking at the data, and plot it against rescaled latitude (grouped by rescaled longitude, see Figure 4.3), against rescaled longitude (grouped by rescaled latitude), and against elevation. The *c* reading plotted against latitude shows linear trend at lower longitudes that vanishes for higher longitude values, and a similar pattern is found for longitude (not shown). There is a slight hint of a negative relationship with elevation (not shown). These are indicative (but not strongly) of global trend.

We start by fitting a purely investigative model, based on serial correlation and with a null fixed model and examine the residuals, using the following commands

```
########################
# baseline model - purely exploratory = allow for serial correlation in measurements
########################
ec.asr <- asreml(c~1,rcov=~ar1(Time),data=ec.df)
summary(ec.asr)
plot(ec.asr)
# inspect residuals
ec.asr <- update(ec.asr,maxiter=1,aom=T)
temp <- ec.asr$aom$R[,"stdCondRes"]
temp.df <- ec.df
temp.df$et <- temp
subset(temp.df,abs(et)>3)[,c('unit','et')]
```

The residual variance is estimated as 109.8 with a high positive serial correlation of 0.79 (output not shown). The residual plots suggest that either the data is non-normal or outliers are present. We therefore do an update on the *asreml* object to calculate standardized conditional residuals and examine these more closely, creating the following list (abbreviated here) by using an arbitrary (and unjustified) threshold of 3

```
> subset(temp.df,abs(et)>3)[,c('unit','et')]
     unit         et
610   610   3.450247
611   611  -5.867524
858   858   3.293241
924   924  -3.113810
1115 1115  -3.565046
1178 1178   4.472470
1179 1179  -4.661485
1252 1252  -3.142963
1308 1308  -5.131899
1309 1309   5.521920
1315 1315  -6.050504
1317 1317   3.847772
1321 1321   3.122365
1574 1574   3.319052
1575 1575  -7.945353
1576 1576   5.654839
1577 1577  -3.592859
...
```

On examining this list, we notice a lot of runs of residuals on successive measurements which have opposing signs. This is unusual when considered in the context of high positive serial correlation estimated across the data set as a whole and therefore perhaps suggestive of an intermittent malfunction in the measurement process, and so we investigate further. We classify areas around the runs plus individual outliers as suspicious and plot these positions, as shown in Figure 4.4. It appears that the large residuals are almost exclusively associated with the motorbike turning process. It is possible (perhaps likely) that the turning process affects the measurement process, and so (in the absence of further information) we wish to also exclude these segments. This

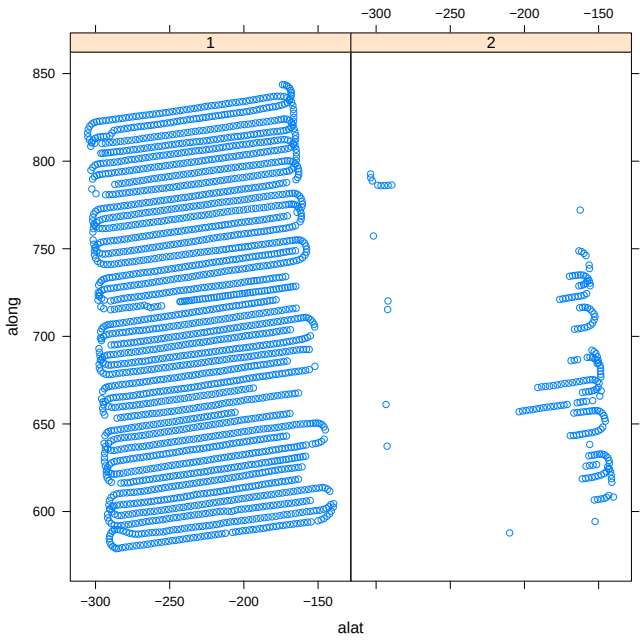

Figure 4.4: Spatial positions, separated by size of standardized conditional residuals (right = large residuals).

is not simple, so we use an approximate calculation based on the expectation that the turning process involves relatively small changes in longitude and relatively large changes in latitude between successive measurements. The resulting partition is shown in Figure 4.5 and has been reasonably (but not completely) successful.

We can exclude the turns and repeat the same process. The residual variance has dropped to 88.6 and the serial correlation has increased to 0.85, but there still seem to be a lot of large residuals and again, the set of large residuals includes several sets of runs containing successive residuals of opposite sign. We again exclude these suspect points and runs, slightly lengthening the runs where these are present. Plotting the residuals thus identified shows that we have picked up a few turns that had been missed, and an additional few runs of suspicious points within the data set (not shown). We will exclude all of these points prior to analysis (giving dataframe *new3ec.df*).

Now the data is in reasonable shape we can start to think about building a model. First, we take a slightly controversial step: we thin the data by taking only every third point. We do this here in order to save computing time, and we can also use the excluded points for model validation. The first reason is a poor one, but the fitting of large geostatistical models using ASReml can be very slow, and this is an area of research for the future. Here we cheat for purposes of easy illustration. The thinned data is held in data frame *new4ec.df*.

We start by forming a directional variogram from the raw data, using the commands below, producing the plot shown in Figure 4.6.

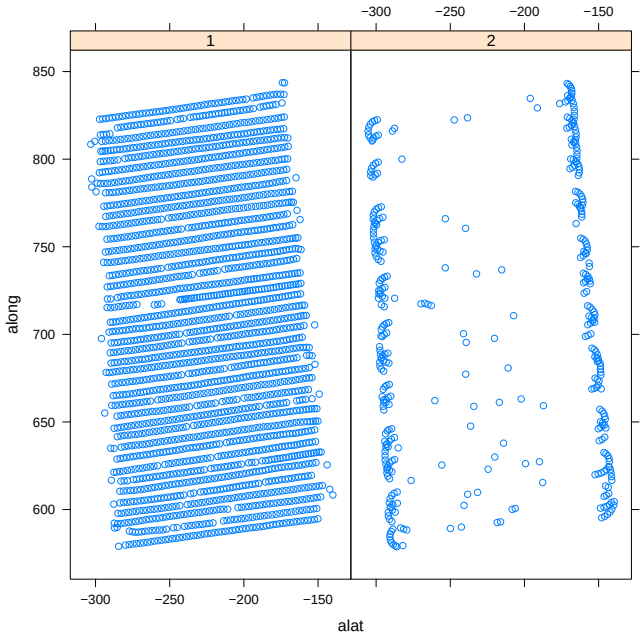

Figure 4.5: Spatial positions, separated by positions identified as turns (right) or not (left).

```
#########################
# null model
#########################
ec.null0.asr <- asreml(c~1,data=new4ec.df)
summary(ec.null0.asr)
#########################
# directional variogram
#########################
ec.null0.vard <- asreml.variogram(y=new4ec.df$alat,x=new4ec.df$along,
        z=residuals(ec.null0.asr),angle=c(0,45,90,135),angle.tol=45)
ec.null0.vard
ec.null0.vard$dir <- as.factor(ec.null0.vard$angle)
xyplot(gamma~distance,group=dir,data=ec.null0.vard,ylab='semi-variance',auto.key=T,type="o")
```

This variogram suggests the presence of non-stationarity in the segments centered on 45° and 90°. In combination with our previous observations of trend in the c reading (eg. Figure 4.3), this reinforces the suggestion of global trend, and we will try to remove this using linear terms in (rescaled) latitude and longitude, allowing for an interaction. The model is fitted and the directional variogram recalculated using the commands below (shown in Figure 4.7).

```
#########################
# use linear surface to remove global trend in certain directions
#########################
ec.null.asr <- asreml(c~1+alat*along,data=new4ec.df)
wald(ec.null.asr, denDF="default")
summary(ec.null.asr)
```

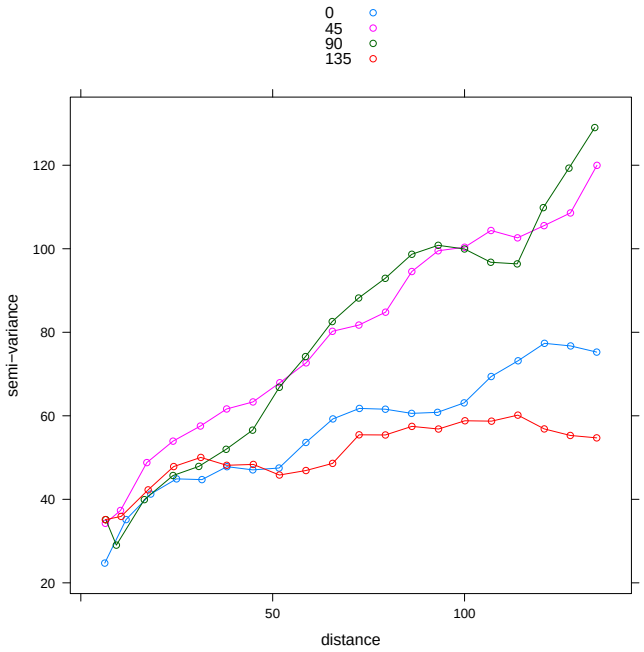

Figure 4.6: Directional variogram calculated from raw (thinned) data frame new4ec.df.

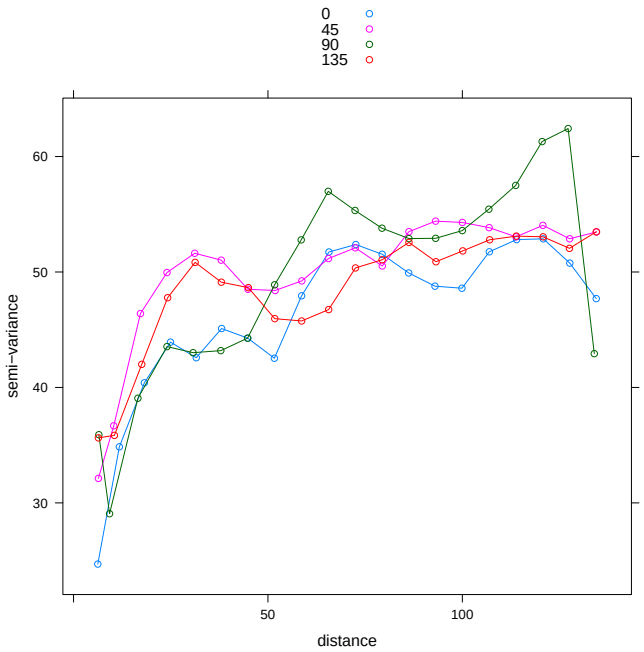

Figure 4.7: Directional variogram calculated from raw (thinned) data frame new4ec.df after removing planar surface (linear x linear).

```
plot(ec.null.asr)
########################
# check whether variogram is better?
########################
ec.null.vard <- asreml.variogram(y=new4ec.df$alat,x=new4ec.df$along,
        z=residuals(ec.null.asr),angle=c(0,45,90,135),angle.tol=45)
ec.null.vard$dir <- as.factor(ec.null.vard$angle)
xyplot(gamma~distance,group=dir,data=ec.null.vard,ylab='semi-variance',auto.key=T,type="o")
```

There is no further sign of non-stationarity, and no indication of anisotropy, as the lines for the four directions cluster together. We will now start to investigate the spatial trend in the data set, using this planar adjustment for global spatial trend. As stated previously, we will use Matérn correlation functions, and will profile over values of ν to avoid issues with estimation of this parameter. Table 4.1 shows a summary of the models fitted to the thinned data set, with their log-likelihood values. All of the models shown here were isotropic. Attempts to fit anisotropic models were unsuccessful, which was not surprising in the context of the directional variograms in Figure 4.7 and so this path was not pursued further. The first set of models (M1-M4) were fitted without a nugget effect, and the second set (M5-M7) with a nugget effect.

Models with a nugget effect were fitted with the nugget as a random term (G-structure) and the spatial term in the R-structure, eg.

```
ec.mat5.asr <- asreml(c~1+alat*along,random=~units,rcov=~mtrn(alat,along,phi=9),
  data=new4ec.df,workspace=8e+08,maxiter=20)
summary(ec.mat5.asr)
```

In this formulation, the specified R-structure is a correlation structure and so θ is automatically added to the model, with estimation performed on the ratio scale. The output from these commands are shown below.

```
> ec.mat5.asr <- asreml(c~1+alat*along,random=~units,rcov=~mtrn(alat,along,phi=9),
+   data=new4ec.df,workspace=8e+08,maxiter=20)

asreml 3.0-1 (1 November 2012), Library: 3.0hj (15 November 2011), X86_64
     LogLik          S2      DF      wall     cpu
  -1582.8407     45.7793   681  09:24:51    10.7
...
  -1581.1233     47.6182   681  09:26:07    13.2
  -1581.1233     47.6224   681  09:26:20    13.2

Finished on: Fri Nov 30 09:26:20 2012

LogLikelihood Converged
> summary(ec.mat5.asr)
$call
asreml(fixed = c ~ 1 + alat * along, random = ~units, rcov = ~mtrn(alat,
    along, phi = 9), data = new4ec.df, workspace = 8e+08, maxiter = 20)
```

```
$loglik
[1] -1581.123

$nedf
[1] 681

$sigma
[1] 6.900896

$varcomp
                     gamma component std.error    z.ratio constraint
units!units.var  0.216922  10.33034  3.671285   2.813820   Positive
R!variance       1.000000  47.62237  7.494053   6.354688   Positive
R!phi           14.087142 670.86303  3.635420 184.535205   Positive
R!nu             0.500000  23.81118        NA         NA      Fixed
R!delta          1.000000  47.62237        NA         NA      Fixed
R!alpha          0.000000   0.00000        NA         NA      Fixed
R!lambda         2.000000  95.24474        NA         NA      Fixed

attr(,"class")
[1] "summary.asreml"
```

For the Matérn model, the summary output of variance parameters is particularly confusing, as they are all re-scaled by the residual variance (θ = *R!variance* = 47.62). In fact, all of the Matérn parameters defining the correlation pattern (ie. ϕ, ν, δ, α, λ) are scale-independent, and take the values given in the *gamma* column. This is a bug in the output that will be fixed - the parameter values stored internally are all correct. To avoid this issue, we could have used the *mtrnv* variance model, which includes a common variance, in order to force estimation on the components scale.

Table 4.1: Summary of sequence of isotropic models fitted to the c reading: bold parameters are fixed at the given value. All models include a planar surface in latitude and longitude as fixed terms. NVP is number of estimated variance parameters. MSEP is mean squared error of prediction on 690 points.

Model	$\hat{\nu}$	$\hat{\phi}$	$\hat{\sigma}_s^2$	$\hat{\sigma}_e^2$	ℓ_R	NVP	MSEP
M0	-	-	-	-	-1706.62	1	51.52
M1	**0.5**	9.02	54.36	-	-1583.48	2	-
M2	**1.0**	4.54	51.23	-	-1590.98	2	-
M3	**1.5**	3.31	50.26	-	-1595.56	2	-
M4	**2.0**	2.70	49.80	-	-1598.48	2	-
M5	**0.5**	14.09	47.62	10.33	-1581.12	3	20.17
M6	**1.0**	9.42	38.77	17.98	-1581.19	3	21.79
M7	**1.5**	7.40	35.82	20.33	-1581.31	3	22.80
M8	**2.0**	6.24	34.35	21.41	-1581.45	3	23.43

From Table 4.1, for the models without a nugget effect, it is clear that the Matérn model with $\nu = 0.5$ gives a better fit in terms of log-likelihood. When a nugget effect is added in, the log-likelihood values across the range of Matérn models are very similar. We therefore used a validation procedure to calculate and compare the MSEP (mean squared error of prediction) across these models.

The principle of the procedure is simple: we predict for a set of points not used to fit the model and calculate the mean squared discrepancy between the predictions and the true values. For our prediction set, we chose one of the two thinned thirds of the data set, giving an even coverage of 690 points.

The difficulties come in the practical management of the process. In order for predictions to include the Matérn process and exclude the nugget (regarded here as noise), the Matérn model must be included in the random model and the nugget as the residual. As discussed in Section 4.3, this is possible when (as here) there are no repeat locations. Even so, successful prediction from this model is tricky to specify correctly. Instead, we use the trick of including the locations to be predicted into our model with missing response values. The fitted values for these locations (excluding the residual term) will then be saved (in the sparse coefficients section of the *asreml* object). It is not sensible to add too many missing responses at a time (slows down the estimation process), so we loop over 6 sets of 115 points. As adding missing responses cannot change the estimates, we can save time by setting the initial values to the final estimates and only allowing one iteration of the algorithm.

To get and set initial values, we use the commands

```
# get estimates to use as initial values in validation
ec.mat5.sv <- asreml(c~1+alat*along,random=~mtrnv(alat,along,phi=9),
  data=new4ec.df,workspace=8e+08,maxiter=20,start.values=T)
iv.5 <- ec.mat5.sv$gammas.table
iv.5$Value[1] <- 14.09
iv.5$Value[6] <- 4.62
iv.5$Value[7] <-10.33
iv.5
```

Note the switching of the Matérn function into the random model formula requires the use of function *mtrnv*, to ensure that a common variance is fitted. There is no need to specify the residual term, which defaults to *id(units)*, with estimation on the ratio scale. Use of the *asreml* function with *start.values=T* returns a list object (here named *ec.mat5.sv*) with three components. The third component, *gammas.table*, is a data frame containing the parameter names, their initial values and boundary constraints. If we set the initial values as required, we can pass these into the *asreml* call using the *G.param* argument. The following commands can then be used to get the predictions and calculate the MSEP.

```
# record of included points and set of points for validation
include.points <- subset(new3ec.df,q==0 & !is.na(new3ec.df$c) )$unit
check.points <- subset(new3ec.df,q==1 & !is.na(new3ec.df$c) )$unit
```

```
# validation for 690 points (even spread)
mvest.5 <- rep(0,690)
cval.5 <- rep(0,690)
for (i in 1:6) {
  l <- 115*(i-1)+1
  h <- 115*i
  new.points <- c(check.points[l:h])
  include <- c(include.points,new.points)
  temp.df <- subset(new3ec.df,is.element(new3ec.df$unit,include))
  temp.df$newc <- temp.df$c
  temp.df$newc[is.element(temp.df$unit,new.points)] <- NA
  ec.temp.asr <- asreml(newc~1+alat*along,random=~mtrnv(alat,along,phi=14),
    data=temp.df,workspace=8e+08,maxiter=1,G.param=iv.5)
  mvest.5[l:h] <- coef(ec.temp.asr)$sparse
  cval.5[l:h] <- subset(new3ec.df,is.element(new3ec.df$unit,new.points))$c
}
mvest.5 <- -1*mvest.5
msep.5 <- sum((cval.5-mvest.5)^2)/length(cval.5)
msep.5
```

The results are shown in Table 4.1, and it appears that the very small difference in log-likelihood values are amplified in the MSEP: the Matérn model with $\nu = 0.5$ gives the most accurate predictions. The directional variograms still look ok, as do residual plots. This fitted model is equivalent to an exponential power model with correlation function $\exp(-d/\phi) = \exp(-1/\phi)^d$. Here $\phi = 14.09$, so we have correlation at a distance of 1 (rescaled) unit of 0.93.

We can predict across a grid to get a picture of the smooth surface. Again, this would be better done with predict (to obtain SEs), but here we have used the missing observations trick again (code below). The resulting grid of predictions is shown in Figure 4.8.

```
# prediction on grid
# set up data frame for prediction
temp.df <- data.frame(alat=new4ec.df$alat,along=new4ec.df$along, c=new4ec.df$c)
nlat <- seq(-300,-140,10)
nla <- length(nlat)
nlong <- seq(580,840,20)
nlo <- length(nlong)
newlat  <- rep(nlat,times=nlo)
newlong <- rep(nlong,each=nla)
pred.df<- data.frame( alat=newlat, along=newlong, c=rep(NA,nlo*nla) )
pred.df
temp.df <- rbind(temp.df,pred.df)
# do prediction and plot results
ec.temp.asr <- asreml(c~1+alat*along,random=~mtrnv(alat,along,phi=14),
  data=temp.df,workspace=8e+08,maxiter=1,G.param=iv.5)
pred.grid <- -1*coef(ec.temp.asr)$sparse
length(pred.grid)
levelplot(pred.grid~newlong*newlat)
```

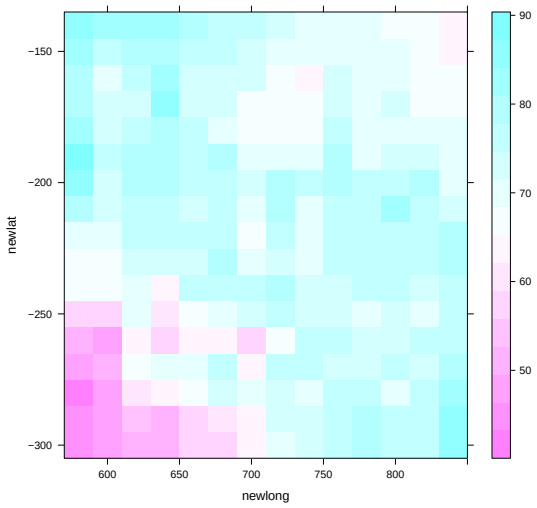

Figure 4.8: Predicted surface from final model.

So, we seem to have a reasonable model for the spatial variation present, but there remains a question over the first 156 measurements, the turns and the odd runs within the area. In practice, we would want to investigate whether these observations are real, implying some sort of discontinuity in the spatial surface, or whether there are issues with the measurement process.

Bibliography

Abramowitz, M. & Stegun, I. A., editors (1965). *Handbook of Mathematical Functions*. Dover Publications, New York.

Breslow, N. & Clayton, D. (1993). Approximate inference in generalised linear mixed models. *Journal of the American Statistical Association* **88**, 9–25.

Brien, C. J. & Demetrio (2009). Formulating mixed models for experiments, including longitudinal experiments. *Journal of Agricultural, Biological and Environmental Statistics* **14**, 253–280.

Butler, D. G., R., C. B., R., G. A., & J., G. B. (2009). ASReml-R reference manual. Technical report, Queensland Department of Primary Industries and fisheries.

Crainiceanu, C. & Ruppert, D. (2004). Likelihood ratio tests in linear mixed models with one variance component. *Journal of the Royal Statistical Society, Series B* **66**, 165–185.

Cullis, B. R. & Gleeson, A. C. (1991). Spatial analysis of field experiments - an extension to two dimensions. *Biometrics* **47**, 1449–1460.

Diggle, P. J., Ribeiro, P. J. J., & Christensen, O. F. (2003). An introduction to model-based geostatistics. In Moller, J., editor, *Spatial Statistics and Computational Methods*, pages 43–86. Springer-Verlag.

Diggle, P. J., Tawn, J. A., & Moyeed, R. A. (1998). Model-based geostatistics (with discussion). *Applied Statistics* **47**, 299–350.

Gillois, M. (1964). La relation didentite en genetique. *Ann Inst Henri Poincar* **B 2**, 1–94.

Gilmour, A. R., Gogel, B. J., Cullis, B. R., & Thompson, R. (2009). *ASREML User Guide, Release 3.0*. VSN International Ltd, 5 The Waterhouse, Waterhouse St, Hemel Hempstead, UK HP1 1ES.

Harris, D. (1964). Genotypic covariances between inbred relatives. *Genetics* **50**, 1319–1348.

Haskard, K. A. (2005). *Anisotropic Matérn correlation and other issues in model-based geostatistics*. PhD thesis, BiometricsSA, University of Adelaide.

Haskard, K. A., Cullis, B. R., & Verbyla, A. P. (2007). Anisotropic matérn correlation and spatial prediction using REML. *Journal of Agricultural and Biological Sciences* **12**, 147–160.

Kammann, E. E. & Wand, M. P. (2003). Geoadditive models. *Applied Statistics* **52**, 1–18.

Kenward, M. G. & Roger, J. H. (1997). The precision of fixed effects estimates from restricted maximum likelihood. *Biometrics* **53**, 983–997.

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa* **52**, 119–139.

Lynch, M. & Walsh, B. (1998). *Genetics and analysis of quantitative traits*. Sinauer Associates.

Maki-Tanila, A. (2007). An overview on quantitative and genomic tools for utilising dominance genetic variation in improving animal production. *Agricultural and Food Science 16:188198* **16**, 188–198.

Martin, R. J. (1979). A subclass of lattice processes applied to a problem in planar sampling. *Biometrika* **66**, 209–217.

Mrode, R. (1995). *Linear models for the prediction of animal breeding values*. CABI Publishing.

Oakey, H., Verbyla, A., Cullis, B., Pitchford, W., & Kuchel, H. (2006). Joint modelling of additive and non-additive genetic line effects in single field trials. *Theoretical and Applied Genetics* **113**, 809–819.

Oakey, H., Verbyla, A., Cullis, B., Wei, X., & Pitchford, W. (2007). Joint modelling of additive and non-additive (genetic line) effects in multi-environment trials. *Theoretical and Applied Genetics* **114**, 1319–1332.

Ripley, B. D. (1981). *Spatial Statistics*. John Wiley, New York.

Smith, A., Cullis, B. R., & Thompson, R. (2001). Analyzing variety by environment data using multiplicative mixed models and adjustments for spatial field trend. *Biometrics* **57**, 1138–1147.

Smith, A. B. (1999). *Multiplicative mixed models for the analysis of multi-environment trial data*. PhD thesis, University of Adelaide.

VSN International (2012). *GenStat for Windows 15th Edition*. VSN International Ltd, 5 The Waterhouse, Waterhouse St, Hemel Hempstead, UK HP1 1ES.

Webster, R. & Oliver, M. A. (2001). *Geostatistics for Environmental Scientists*. John Wiley and Sons, Chichester.

Welham, S., Cullis, B. R., Gogel, B. J., Gilmour, A. R., & Thompson, R. (2004). Prediction in linear mixed models. *Australian and New Zealand Journal of Statistics* **46**, 325–347.

Welham, S. J. & Thompson, R. (1997). Likelihood ratio tests for fixed model terms using residual maximum likelihood. *JRSS(B)* **59**, 701–714.

White, I., Rainer, R. Knap, P., & Brotherstone, S. (2006). Variance components for survival of piglets at farrowing using a reduced animal model. *Genetic Selection and Evolution* **38**, 359–370.

Wilkinson, G. N. & Rogers, C. E. (1973). Symbolic description of factorial models for analysis of variance. *Applied Statistics* **22**, 392–399.