
ASReml User Guide

Release 1.0

A R Gilmour
NSW Agriculture, Orange, Australia

B J Gogel
Department of Primary Industries, Brisbane, Australia

B R Cullis
NSW Agriculture, Wagga Wagga, Australia

S J Welham
Rothamsted Research, Harpenden, United Kingdom

R Thompson
Rothamsted Research, Harpenden, United Kingdom

ASReml User Guide Release 1.0

A R Gilmour, B J Gogel, B R Cullis, S J Welham and R Thompson

Published by:

VSN International Ltd,
5 The Waterhouse,
Waterhouse Street,
Hemel Hempstead,
HP1 1ES, UK
E-mail: info@vsn-intl.com
Website: <http://www.vsn-intl.com/>

Copyright Notice

Copyright © 2002, NSW Agriculture. All rights reserved.

Except as permitted under the Copyright Act 1968 (Commonwealth of Australia), no part of the publication may be reproduced by any process, electronic or otherwise, without specific written permission of the copyright owner. Neither may information be stored electronically in any form whatever without such permission.

The correct bibliographical reference for this document is:

Gilmour, A.R., Gogel, B.J., Cullis, B.R., Welham, S.J. and Thompson, R. 2002 ASReml User Guide Release 1.0 VSN International Ltd, Hemel Hempstead, HP1 1ES, UK

ISBN 1-904375-07-3

Author email addresses

Arthur.Gilmour@agric.nsw.gov.au
Beverley.Gogel@dpi.qld.gov.au
Brian.Cullis@agric.nsw.gov.au
Sue.Welham@bbsrc.ac.uk
Robin.Thompson@bbsrc.ac.uk

Preface

ASReml is a statistical package that fits linear mixed models using Residual Maximum Likelihood (REML). It has been under development since 1993 and is a joint venture between the Biometrics Program of NSW Agriculture and the Biomathematics Unit previously the Statistics Department of Rothamsted Experiment Station. This guide relates to the August 2002 version of ASReml.

Linear mixed effects models provide a rich and flexible tool for the analysis of many data sets commonly arising in the agricultural, biological, medical and environmental sciences. Typical applications include the analysis of (un)balanced longitudinal data, repeated measures analysis, the analysis of (un)balanced designed experiments, the analysis of multi-environment trials, the analysis of both univariate and multivariate animal breeding and genetics data and the analysis of regular or irregular spatial data.

One of the strengths of ASReml is the wide range of possible variance models for the random effects in the linear mixed model. There is a potential cost for this wide choice. Users should be aware of the dangers of either overfitting or attempting to fit inappropriate variance models to small or highly unbalanced data sets. We stress the importance of the use of data-driven diagnostics and encourage the user to read the examples chapter, in which we have attempted to not only present the syntax of ASReml in the context of real analyses but also to indicate some of the modelling approaches we have found useful.

Another strength of ASReml is the use of the Average Information (AI) algorithm and sparse matrix methods for fitting the linear mixed model. This enables ASReml to analyse large and complex data sets quite efficiently. We have concentrated on expanding the range of variance models and improving the speed and reliability of the *engine* rather than providing comprehensive reporting facilities. However, the facilities are adequate for most situations. The *engine* of ASReml is used by the REML directive in GENSTAT and by the `samm` class of functions in S-PLUS. Both of these have good data manipulation and graphical facilities.

The guide has 15 chapters. Chapter 1 introduces ASReml and describes the conventions used in this guide. Chapter 2 outlines some basic theory while Chapter 3 presents an overview of the syntax of ASReml through a simple example. Data file preparation is described in Chapter 4 and Chapter 5 describes how to input data into ASReml. Chapters 6 and 7 are key chapters which present the syntax for specifying the linear model and the variance models for the random effects in the linear mixed model. Chapters 8 and 9 describe special commands for multivariate and genetic analyses respectively. Chapter 10 deals with prediction of linear functions of fixed and random effects in the linear mixed model and Chapter 11 presents the syntax for forming functions of variance components. Chapter 12 demonstrates running an ASReml job and explains the interactive menu features available and Chapter 13 gives a detailed explanation of the output files. Chapter 14 gives an overview of the error messages generated in ASReml and some guidance as to their probable cause. The guide concludes with the most extensive chapter which presents the examples.

The data sets and ASReml input files used in this guide are available for electronic distribution from <http://vsn-intl.com/asrem1>. They remain the property of the authors or of the original source but may be freely distributed provided the source is acknowledged. The authors would appreciate feedback and suggestions for improvements to the program and this guide.

Upgrades

ASReml is being continually upgraded to implement new developments in the application of linear mixed models. The release version will be distributed by VSN on CD to licensed users. A developmental version is available to licensees of the current release version via the ASReml website. This User Guide and a Reference Manual are also available on that site. Most users will not need to access the developmental version unless they are actively involved in testing a new development.

Acknowledgements

We gratefully acknowledge the Grains Research and Development Corporation of Australia for their financial support for our research since 1988. Brian Cullis and Arthur Gilmour wish to thank NSW Agriculture, and in particular Alan Gleeson for having faith in the project and for providing a stimulating and exciting environment for applied biometrical research and consulting. Rothamsted Research receives grant-aided support from the Biotechnology and Biological Sciences Research Council of the United Kingdom.

We sincerely thank Ari Verbyla, Dave Butler and Alison Smith, the other members of the ASReml ‘team’. Ari is thanked for his contribution to the cubic smoothing splines technology, on-going testing of the software and numerous helpful discussions and insight. Dave is thanked for his support of the project and in the latter years the development of the `samm` class of functions. Alison is thanked for her contribution to Chapter 15, the development of many of the approaches for the analysis of multi-section trials, and constant and undeterred commitment to ASReml. We also thank Ian White for his contribution to the spline methodology and Simon Harding and Megan Cherry for licensing and installation software.

ASReml was available free of charge for a number of years and we acknowledge our users who have supported the package through its early development. We look forward to their continued support and interaction. ASReml is now distributed by VSN-International on behalf of NSW Agriculture and Rothamsted Research. Proceeds are used to support continued development. Please visit <http://www.VSN-Intl.com> or email info@VSN-Intl.com for further information.

Arthur Gilmour acknowledges the grace of God through Jesus Christ who enables this research to proceed. *Be exalted O God, above the heavens: and Thy glory above all the earth. Psalm 108:5.*

Contents

Preface	i
List of Tables	xvi
List of Figures	xviii
1 Introduction	1
1.1 What ASReml can do	2
1.2 Installation	2
1.3 How to use this guide	3
1.4 Help and discussion list	3
1.5 Typographic conventions	4
2 Some theory	5
2.1 The linear mixed model	6
Introduction	6
Direct product structures	6
Variance structures for the errors: R structures	8

Variance structures for the random effects: G structures	9
2.2 Estimation	10
Estimation of the variance parameters	10
Estimation/prediction of the fixed and random effects	13
2.3 What are BLUPs?	14
2.4 Combining variance models	15
2.5 Inference	16
Tests of hypotheses: variance parameters	16
Diagnostics	17
Tests of hypotheses: fixed effects	18
3 A guided tour	19
3.1 Introduction	20
3.2 Nebraska Intrastate Nursery (NIN) field experiment	20
3.3 The ASReml data file	22
3.4 The ASReml command file	23
The title line	24
Reading the data	24
The data file line	25
Specifying the terms in the mixed model	25
Tabulation	26

Prediction	26
Variance structures	26
3.5 Running the job	27
3.6 Description of output files	27
The .asr file	27
The .sln file	29
The .yht file	30
3.7 Tabulation, predicted values and functions of the variance components	31
4 Data file preparation	33
4.1 Introduction	34
4.2 The data file	34
Free format data files	34
Fixed format data files	36
Binary format data files	36
5 Command file: Reading the data	37
5.1 Introduction	38
5.2 Important rules	38
5.3 Title line	39
5.4 Specifying and reading the data	39
Data field definition syntax	39

5.5	Transforming the data	41
	Transformation syntax	41
	Other rules and examples	46
	Special note on covariates	47
5.6	Data file line	47
	Data line syntax	47
5.7	Data file qualifiers	48
5.8	Job control qualifiers	49
6	Command file: Specifying the terms in the mixed model	61
6.1	Introduction	62
6.2	Specifying model formulae in ASReml	62
	General rules	62
	Examples	66
6.3	Fixed terms in the model	66
	Primary fixed terms	66
	Sparse fixed terms	67
6.4	Random terms in the model	67
6.5	Interactions and conditional factors	68
	Interactions	68
	Conditional factors	68

6.6	Alphabetic list of model functions	69
6.7	Weights	72
6.8	Missing values	73
	Missing values in the response	73
	Missing values in the explanatory variables	73
6.9	Some technical details about model fitting in ASReml	74
	Sparse <i>versus</i> dense	74
	Ordering of terms in ASReml	74
	Aliasing and singularities	74
	Examples of aliasing	75
6.10	Testing of terms	76
7	Command file: Specifying the variance structures	77
7.1	Introduction	78
	Non singular variance matrices	78
7.2	Variance model specification in ASReml	79
7.3	A sequence of structures for the NIN data	79
7.4	Variance structures	87
	General syntax	87
	Variance header line	89
	R structure definition	90

G structure header and definition lines	92
7.5 Variance model description	92
Forming variance models from correlation models	93
Additional notes on variance models	94
7.6 Variance structure qualifiers	103
7.7 Rules for combining variance models	104
7.8 G structures involving more than one random term	105
7.9 Constraining variance parameters	106
Parameter constraints within a variance model	106
Constraints between and within variance models	107
7.10 Model building using the !CONTINUE qualifier	109
8 Command file: Multivariate analysis	110
8.1 Introduction	111
Wether trial data	111
8.2 Model specification	112
8.3 Variance structures	113
Specifying multivariate variance structures in ASReml	113
8.4 The output for a multivariate analysis	114
8.5 More than two response variates	116
9 Command file: Genetic analysis	117

9.1	Introduction	118
9.2	The command file	118
9.3	The pedigree file	119
9.4	Reading in the pedigree file	120
9.5	Genetic groups	121
9.6	Reading a user defined inverse relationship matrix	123
	The example continued	124
10	Tabulation of the data and prediction from the model	125
10.1	Introduction	126
10.2	Tabulation	128
10.3	Prediction	129
	Examples	133
11	Functions of variance components	134
11.1	Introduction	135
11.2	Syntax	135
	Linear combinations of components	135
	Heritability	136
	Correlation	137
	A more detailed example	138

12 Command file: Running the job	140
12.1 Introduction	141
12.2 The command line	141
12.3 Command line options	142
Debug command line options (D, E)	143
Graphics command line options (G, I, N)	143
Job control command line options (C, F, R)	144
Memory command line options (S)	144
Menu command line options (M)	145
Log file (L)	146
Post-processing calculation of variance components (P)	146
Examples	147
12.4 Advanced processing arguments	147
Standard use of arguments	147
Recycle	148
Entering command line arguments interactively	148
13 Description of output files	149
13.1 Introduction	150
13.2 An example	151
13.3 Key output files	151

The .asr file	151
The .sln file	154
The .yht file	155
13.4 Other ASReml output files	156
The .res file	156
The .vrb file	159
The .vvp file	161
The .rsv file	162
The .dpr file	162
The .pvc file	162
The .pvs file	162
The .tab file	164
13.5 ASReml output objects and where to find them	165
14 Error messages	168
14.1 Introduction	169
14.2 Common problems	170
14.3 Things to check in the .asr file	171
14.4 An example	173
14.5 Information, Warning and Error messages	184
15 Examples	193

15.1 Introduction	194
15.2 Split plot design - Oats	194
15.3 Unbalanced nested design - Rats	198
15.4 Source of variability in unbalanced data - Volts	202
15.5 Balanced repeated measures - Height	204
15.6 Spatial analysis of a field experiment - Barley	213
15.7 Unreplicated early generation variety trial - Wheat	219
15.8 Paired Case-Control study - Rice	225
Standard analysis	226
A multivariate approach	231
Interpretation of results	235
15.9 Balanced longitudinal data - Random coefficients and cubic smoothing splines - Oranges	237
15.10 Multivariate animal genetics data - Sheep	245
Half-sib analysis	246
Animal model	253
Bibliography	257
Index	260

List of Tables

2.1	Combination of models for G and R structures	15
3.1	Trial layout and allocation of varieties to plots in the NIN field trial .	21
5.1	List of transformation qualifiers and their actions with examples . . .	43
5.2	Qualifiers relating to data input and output	48
5.3	List of frequently used job control qualifiers	49
5.4	List of occasionally used job control qualifiers	50
5.5	List of rarely used job control qualifiers	55
5.6	List of very rarely used job control qualifiers	59
6.1	Summary of reserved words, allowable operators and functions	64
6.2	Alphabetic list of model functions and descriptions	69
6.3	Examples of aliasing in ASReml	75
6.4	Schematic analysis of variance table from fitting the model $yield \sim \mu A B X A.B A.X B.X A.B.X !r rep$	76
7.1	Sequence of variance structures for the NIN field trial data	86
7.2	Schematic outline of variance model specification in ASReml	88

7.3	Details of the variance models available in ASReml	99
7.4	List of R and G structure qualifiers	103
7.5	Examples of constraining variance parameters in ASReml	107
9.1	List of pedigree file qualifiers	121
10.1	List of prediction qualifiers	131
12.1	Command line options	142
12.2	The use of arguments in ASReml	148
13.1	Summary of ASReml output files	150
13.2	ASReml output objects and where to find them	165
14.1	Some information messages and comments	184
14.2	List of warning messages and likely meaning(s)	185
14.3	Alphabetical list of error messages and probable cause(s)	188
15.1	A split-plot field trial of oat varieties and nitrogen application	194
15.2	Rat data: AOV decomposition	199
15.3	Summary of F-tests of fixed effects adjusted for the other factors for the rat data	202
15.4	REML log-likelihood ratio for the variance components in the volt- age data	205
15.5	Summary of variance models fitted to the plant data	206

15.6	Summary of Wald test for fixed effects for variance models fitted to the plant data	213
15.7	Field layout of Slate Hall Farm experiment	214
15.8	Summary of models for the Slate Hall data	219
15.9	Estimated variance components from univariate analyses of bloodworm data. (a) Model with homogeneous variance for all terms and (b) Model with heterogeneous variance for interactions involving tmt	229
15.10	Equivalence of random effects in bivariate and univariate analyses . .	231
15.11	Estimated variance parameters from bivariate analysis of bloodworm data	233
15.12	Orange data: AOV decomposition	241
15.13	Sequence of models fitted to the Orange data	242
15.14	REML estimates of a subset of the variance parameters for each trait for the genetic example, expressed as a ratio to their asymptotic s.e.	247
15.15	Wald tests of the fixed effects for each trait for the genetic example	247
15.16	Variance models fitted for each part of the ASReml job in the analysis of the genetic example	250

List of Figures

13.1	Variogram of residuals	158
13.2	Plot of residuals in field plan order	159
13.3	Plot of the marginal means of the residuals	160
13.4	Histogram of residuals	160
15.1	Residual plot for the rat data	201
15.2	Residual plot for the voltage data	204
15.3	Trellis plot of the height for each of 14 plants	205
15.4	Residual plots for the EXP variance model for the plant data	209
15.5	Sample variogram of the $AR1 \times AR1$ model for the Slate Hall data	216
15.6	Sample variogram of the $AR1 \times AR1$ model for the Tullibigeal data	222
15.7	Sample variogram of the $AR1 \times AR1 + \text{pol}(\text{column}, -1)$ model for the Tullibigeal data	224
15.8	Rice bloodworm data: Plot of square root of root weight for treated versus control	226
15.9	BLUPs for treated for each variety plotted against BLUPs for control	234
15.10	Estimated deviations from regression of treated on control for each variety plotted against estimate for control	235

15.11	Estimated difference between control and treated for each variety plotted against estimate for control	236
15.12	Trellis plot of trunk circumference for each tree	238
15.13	Fitted cubic smoothing spline for tree 1	240
15.14	Plot of fitted cubic smoothing spline for model 1	243
15.15	Trellis plot of trunk circumference for each tree at sample dates (adjusted for <i>season</i> effects), with fitted profiles across time and confidence intervals	244
15.16	Plot of the residuals from the nonlinear model of Pinheiro and Bates	245

1

Introduction

What ASReml can do

Installation

How to use the guide

Help and discussion list

Typographic conventions

1.1 What ASReml can do

ASReml (pronounced *A S Remel*) is used to fit linear mixed models to quite large data sets with complex variance models. It extends the range of variance models available for the analysis of experimental data. ASReml has application in the analysis of

- (un)balanced longitudinal data,
- repeated measures data (multivariate analysis of variance and spline type models),
- (un)balanced designed experiments,
- multi-environment trials and meta analysis,
- univariate and multivariate animal breeding and genetics data (involving a relationship matrix for correlated effects),
- regular or irregular spatial data.

The engine of ASReml forms the basis of the REML procedure in GENSTAT. An interface for SPLUS called `samm` has also been developed. While these interfaces will be adequate for many analyses, some large problems will need to use ASReml. The ASReml user interface is concise. Most effort has been directed towards efficiency of the engine. It normally operates in a batch mode.

Problem size depends on the sparsity of the mixed model equations and the size of your computer. However, models with 100,000 effects have been fitted successfully. The computational efficiency of ASReml arises from using the Average Information REML procedure (giving quadratic convergence) and sparse matrix operations. ASReml has been operational since March 1996 and is updated periodically.

1.2 Installation

Installation instructions vary for the various platforms and are distributed with the program. If you find the instructions inadequate for some reason, please email asrem1@VSN-Intl.com.

1.3 How to use this guide

- Theory** The guide consists of 15 chapters. Chapter 1 introduces ASReml and describes the conventions used in the guide. Chapter 2 outlines some basic theory which you may need to come back to.
- Getting started** New ASReml users are advised to read Chapter 3 before attempting to code their first job. It presents an overview of basic ASReml coding demonstrated on a real data example. Chapter 15 presents a range of examples to assist users further. When coding your first job, look for an example to use as a template.
- Examples**
- Data file** Data file preparation is described in Chapter 4, while Chapter 5 describes how to input data into ASReml. Chapters 6 and 7 are key chapters which present the syntax for specifying the linear model and the variance models for the random effects in the linear mixed model. Variance modelling is a complex aspect of analysis. We introduce variance modelling in ASReml by example in Chapter 7.
- Linear model**
- Variance model**
- Prediction** Chapters 8 and 9 describe special commands for multivariate and genetic analyses respectively. Chapter 10 deals with prediction of fixed and random effects from the linear mixed model and Chapter 11 presents the syntax for forming functions of variance components.
- Output** Chapter 12 discusses running an ASReml job and explains the interactive menu features available. Chapter 13 gives a detailed explanation of the output files. Chapter 14 gives an overview of the error messages generated in ASReml and some guidance as to their probable cause.

1.4 Help and discussion list

Supported users of ASReml may email asrem1@VSN-Int1.com for assistance. When requesting help for a job that is not working as you expect, please send the input command file, the data file and the corresponding primary output file along with a description of the problem.

There is also an ASReml discussion list. To join (leave) the list, send an email to ASRem1-request@chiswick.anprod.csiro.au with the word `subscribe` (`unsubscribe`) in the body of the message.

The address for messages to the list is ASRem1@chiswick.anprod.csiro.au.

1.5 Typographic conventions

A hands on approach is the best way to develop a working understanding of a new computing package. We therefore begin by presenting a guided tour of ASReml using a sample data set for demonstration (see Chapter 3). Throughout the guide new concepts are demonstrated by example wherever possible.

In this guide you will often find framed sample boxes to the right of the page as shown here. These contain ASReml command file (sample) code. Note that

- the code under discussion is highlighted in bold type for easy identification,
- the continuation symbol (`:`) is used to indicate that some of the original code is omitted.

```
An example ASReml code box
bold type highlights sections
of code currently under
discussion
remaining code is not
highlighted
: indicates that some of the
original code is omitted from
the display
```

Data examples are displayed in larger boxes in the body of the text, see, for example, page 34. Other conventions are as follows:

- keyboard key names appear in SMALLCAPS, for example, TAB and ESC,
- example code within the body of the text is in **this size and font** and is highlighted in bold type, see pages 27 and 40,
- in the presentation of general ASReml syntax, for example
`[path] asrem1 basename [.as] [arguments]`
 - **typewriter font** is used for text that must be typed verbatim, for example, `asrem1` and `.as` after *basename* in the example,
 - *italic font* is used to name information to be supplied by the user, for example, *basename* stands for the name of a file with an `.as` filename extension,
 - square brackets indicate that the enclosed text and/or arguments are not always required.
- ASReml output is in **this size and font**, see page 28,
- **this font** is used for all other code.

2

Some theory

The linear mixed model

- Introduction
- Direct product structures
- Variance structures for the errors: R structures
- Variance structures for the random effects: G structures

Estimation

- Estimation of the variance parameters
- Estimation/prediction of fixed and random effects

What are BLUPs?

Combining variance models

Inference

- Tests of hypotheses: variance parameters
- Tests of hypotheses: fixed effects

2.1 The linear mixed model

Introduction

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

$$\mathbf{y} = \mathbf{X}\boldsymbol{\tau} + \mathbf{Z}\mathbf{u} + \mathbf{e} \quad (2.1)$$

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

The model (2.1) is called a linear mixed model or linear mixed effects model. It is assumed

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{e} \end{bmatrix} \sim N \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \theta \begin{bmatrix} \mathbf{G}(\boldsymbol{\gamma}) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}(\boldsymbol{\phi}) \end{bmatrix} \right) \quad (2.2)$$

where the matrices \mathbf{G} and \mathbf{R} are functions of parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$, respectively. The parameter θ is a variance parameter which we will refer to as the scale parameter. In mixed effects models with more than one residual variance, arising for example in the analysis of data with more than one section (see below) or variate, the parameter θ is fixed to one. In mixed effects models with a single residual variance then θ is equal to the residual variance (σ^2). In this case \mathbf{R} must be a correlation matrix (see Table 2.1 for a discussion).

Direct product structures

To undertake variance modelling in ASReml you need to understand the formation of variance structures via direct products (\otimes). The direct product of two matrices $\mathbf{A}^{(m \times p)}$ and $\mathbf{B}^{(n \times q)}$ is

$$\begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1p}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \ddots & a_{mp}\mathbf{B} \end{bmatrix}$$

Direct products in R structures

Consider a vector of common errors associated with an experiment. The usual least squares assumption (and the default in ASReml) is that these are independently and identically distributed (IID). However, if \mathbf{e} was from a field experiment

laid out in a rectangular array of r rows by c columns, we could arrange the residuals as a matrix and might consider that they were autocorrelated within rows and columns. Writing the residuals as a vector in field order, that is, by sorting the residuals rows within columns (plots within blocks) the variance of the residuals might then be

$$\sigma_e^2 \Sigma_c(\rho_c) \otimes \Sigma_r(\rho_r)$$

where $\Sigma_c(\rho_c)$ and $\Sigma_r(\rho_r)$ are correlation matrices for the row model (order r , autocorrelation parameter ρ_r) and column model (order c , autocorrelation parameter ρ_c) respectively. More specifically, a two-dimensional separable autoregressive spatial structure (AR1 \otimes AR1) is sometimes assumed for the common errors in a field trial analysis (see Gogel (1997) and Cullis *et al.* (1998) for examples). In this case

$$\Sigma_r = \begin{bmatrix} 1 & & & & \\ \rho_r & 1 & & & \\ \rho_r^2 & \rho_r & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \rho_r^{r-1} & \rho_r^{r-2} & \rho_r^{r-3} & \dots & 1 \end{bmatrix} \quad \text{and} \quad \Sigma_c = \begin{bmatrix} 1 & & & & \\ \rho_c & 1 & & & \\ \rho_c^2 & \rho_c & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \rho_c^{c-1} & \rho_c^{c-2} & \rho_c^{c-3} & \dots & 1 \end{bmatrix}.$$

See Chapter 8 for further details. Alternatively, the residuals might relate to a multivariate analysis with n_t traits and n units and be ordered traits *within* units. In this case an appropriate variance structure might be

$$\mathbf{I}_n \otimes \Sigma$$

where $\Sigma^{(n_t \times n_t)}$ is a general or *unstructured* variance matrix.

Direct products in G structures

Likewise, the random terms in \mathbf{u} in the model may have a direct product structure. For example, for a field trial with s sites, g varieties and the effects ordered varieties *within* sites, the model term *site.variety* may have the variance structure

$$\Sigma \otimes \mathbf{I}_g$$

where Σ is the variance matrix for sites. This would imply that the varieties are independent random effects within each site, have different variances at each site, and are correlated across sites. **Important** Whenever a random term is formed as the interaction of two factors you should consider whether the IID assumption is sufficient or if a direct product structure might be more appropriate.

Variance structures for the errors: \mathbf{R} structures

The vector \mathbf{e} will in some situations be a series of vectors indexed by a factor or factors. The convention we adopt is to refer to these as *sections*. Thus $\mathbf{e} = [e'_1, e'_2, \dots, e'_s]'$ and the e_j represent the errors of *sections* of the data. For example, these sections may represent different experiments in a multi-environment trial (MET), or different trials in a meta analysis. It is assumed that \mathbf{R} is the direct sum of s matrices \mathbf{R}_j , $j = 1 \dots s$, that is,

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

so that each section has its own variance structure which is assumed to be independent of the structures in other sections.

A structure for the residual variance for the spatial analysis of multi-environment trials (Cullis *et al.*, 1998) is given by

$$\begin{aligned} \mathbf{R}_j &= \mathbf{R}_j(\phi_j) \\ &= \sigma_j^2(\boldsymbol{\Sigma}_j(\boldsymbol{\rho}_j)). \end{aligned}$$

Each section represents a trial and we note that this model accounts for between trial error variance heterogeneity (σ_j^2) and possibly a different spatial variance model for each trial.

In the simplest case the matrix \mathbf{R} could be known and proportional to an identity matrix. Each component matrix, \mathbf{R}_j (or \mathbf{R} itself for one section) is assumed to be the direct product (see Searle, 1982) of one, two or three component matrices. The component matrices are related to the underlying structure of the data. If the structure is defined by factors, for example, replicates, rows and columns, then the matrix \mathbf{R} can be constructed as a direct product of three matrices describing the nature of the correlation across replicates, rows and columns. These factors must completely describe the structure of the data, which means that

1. the number of combined levels of the factors must equal the number of data points,
2. each factor combination must uniquely specify a single data point.

These conditions are necessary to ensure the expression $\text{var}(\mathbf{e}) = \boldsymbol{\theta}\mathbf{R}$ is valid. The assumption that the overall variance structure can be constructed as a direct

product of matrices corresponding to underlying factors is called the assumption of separability and assumes that any correlation process across levels of a factor is independent of any other factors in the term. Multivariate data and repeated measures data usually satisfy the assumption of separability. In particular, if the data are indexed by factors **units** and **traits** (for multivariate data) or **times** (for repeated measures data), then the R structure may be written as $units \otimes traits$ or $units \otimes times$. This assumption is sometimes required to make the estimation process computationally feasible, though it can be relaxed, for certain applications, for example fitting isotropic covariance models to irregularly spaced spatial data.

Variance structures for the random effects: **G** structures

The $q \times 1$ vector of random effects is often composed of b subvectors $\mathbf{u} = [\mathbf{u}'_1 \ \mathbf{u}'_2 \ \dots \ \mathbf{u}'_b]'$ where the subvectors \mathbf{u}_i are of length q_i and these subvectors are usually assumed independent normally distributed with variance matrices $\theta \mathbf{G}_i$. Thus just like **R** we have

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

There is a corresponding partition in **Z**, $\mathbf{Z} = [\mathbf{Z}_1 \ \mathbf{Z}_2 \ \dots \ \mathbf{Z}_b]$. As before each submatrix, \mathbf{G}_i , is assumed to be the direct product of one, two or three component matrices. These matrices are indexed for each of the factors constituting the term in the linear model. For example, the term *site.genotype* has two factors and so the matrix \mathbf{G}_i is comprised of two component matrices defining the variance structure for each factor in the term.

Models for the component matrices \mathbf{G}_i include the standard model for which $\mathbf{G}_i = \gamma_i \mathbf{I}_{q_i}$ and direct product models for correlated random factors given by

$$\mathbf{G}_i = \mathbf{G}_{i1} \otimes \mathbf{G}_{i2} \otimes \mathbf{G}_{i3}$$

for three component factors. The vector \mathbf{u}_i is therefore assumed to be the vector representation of a 3-way array. For two factors the vector \mathbf{u}_i is simply the vec of a matrix with rows and columns indexed by the component factors in the term, where vec of a matrix is a function which stacks the columns of its matrix argument below each other.

A range of models are available for the components of both **R** and **G**. They include correlation (*C*) models (that is, where the diagonals are 1), or covariance

(V) models and are discussed in detail in Chapter 7. Some correlation models include

- autoregressive (order 1 or 2)
- moving average (order 1 or 2)
- ARMA(1,1)
- uniform
- banded
- general correlation.

Some of the covariance models include

- diagonal (that is, independent with heterogeneous variances)
- antedependence
- unstructured
- factor analytic.

There is the facility within ASReMl to allow for a nonzero covariance between the subvectors of \mathbf{u} , for example in random regression models. In this setting the intercept and say the slope for each unit are assumed to be correlated and it is more natural to consider the the two component terms as a single term, which gives rise to a single G structure. This concept is discussed later.

2.2 Estimation

Estimation involves two processes that are closely linked. They are performed within the ‘engine’ of ASReMl. One process involves estimation of $\boldsymbol{\tau}$ and prediction of \mathbf{u} (although the latter may not always be of interest) for given θ , $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$. The other process involves estimation of these variance parameters. Note that in the following sections we have set $\theta = 1$ to simplify the presentation of results.

Estimation of the variance parameters

Estimation of the variance parameters is carried out using residual or restricted maximum likelihood (REML), developed by Patterson and Thompson (1971). An historical development of the theory can be found in Searle *et al.* (1992). Note firstly that

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\tau}, \mathbf{H}). \quad (2.3)$$

where $\mathbf{H} = \mathbf{R} + \mathbf{ZGZ}'$. REML does not use (2.3) for estimation of variance parameters, but rather uses a distribution free of $\boldsymbol{\tau}$, essentially based on error contrasts or *residuals*. The derivation given below is presented in Verbyla (1990).

We transform \mathbf{y} using a non-singular matrix $\mathbf{L} = [\mathbf{L}_1 \ \mathbf{L}_2]$ such that

$$\mathbf{L}'_1 \mathbf{X} = \mathbf{I}_p, \quad \mathbf{L}'_2 \mathbf{X} = \mathbf{0}.$$

If $\mathbf{y}_j = \mathbf{L}'_j \mathbf{y}$, $j = 1, 2$,

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim N \left(\begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{L}'_1 \mathbf{H} \mathbf{L}_1 & \mathbf{L}'_1 \mathbf{H} \mathbf{L}_2 \\ \mathbf{L}'_2 \mathbf{H} \mathbf{L}_1 & \mathbf{L}'_2 \mathbf{H} \mathbf{L}_2 \end{bmatrix} \right).$$

The full distribution of $\mathbf{L}'\mathbf{y}$ can be partitioned into a *conditional distribution*, namely $\mathbf{y}_1 | \mathbf{y}_2$, for estimation of $\boldsymbol{\tau}$, and a *marginal distribution* based on \mathbf{y}_2 for estimation of $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$; the latter is the basis of the **residual likelihood**.

The estimate of $\boldsymbol{\tau}$ is found by equating \mathbf{y}_1 to its conditional expectation, and after some algebra we find,

$$\hat{\boldsymbol{\tau}} = (\mathbf{X}'\mathbf{H}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{H}^{-1}\mathbf{y}$$

Estimation of $\boldsymbol{\kappa} = [\boldsymbol{\gamma}' \ \boldsymbol{\phi}']'$ is based on the log residual likelihood,

$$\begin{aligned} \ell_R &= -\frac{1}{2}(\log \det \mathbf{L}'_2 \mathbf{H}^{-1} \mathbf{L}_2 + \mathbf{y}'_2 (\mathbf{L}'_2 \mathbf{H} \mathbf{L}_2)^{-1} \mathbf{y}_2) \\ &= -\frac{1}{2}(\log \det \mathbf{X}'\mathbf{H}^{-1}\mathbf{X} + \log \det \mathbf{H} + \mathbf{y}'\mathbf{P}\mathbf{y}_2) \end{aligned} \quad (2.4)$$

where

$$\mathbf{P} = \mathbf{H}^{-1} - \mathbf{H}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{H}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{H}^{-1}.$$

Note that $\mathbf{y}'\mathbf{P}\mathbf{y} = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\tau}})' \mathbf{H}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\tau}})$. The log-likelihood (2.4) depends on \mathbf{X} and not on the particular non-unique transformation defined by \mathbf{L} .

The log residual likelihood (ignoring constants) can be written as

$$\ell_R = -\frac{1}{2}(\log \det \mathbf{C} + \log \det \mathbf{R} + \log \det \mathbf{G} + \mathbf{y}'\mathbf{P}\mathbf{y}). \quad (2.5)$$

We can also write

$$\mathbf{P} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1}$$

with $\mathbf{W} = [\mathbf{X} \ \mathbf{Z}]$. Letting $\boldsymbol{\kappa} = (\boldsymbol{\gamma}, \boldsymbol{\phi})$, the REML estimates of κ_i are found by calculating the score

$$U(\kappa_i) = \partial \ell_R / \partial \kappa_i = -\frac{1}{2} [\text{tr}(\mathbf{P}\mathbf{H}_i) - \mathbf{y}'\mathbf{P}\mathbf{H}_i\mathbf{P}\mathbf{y}] \quad (2.6)$$

and equating to zero. Note that $\mathbf{H}_i = \partial \mathbf{H} / \partial \kappa_i$.

The elements of the observed information matrix are

$$\begin{aligned} -\frac{\partial^2 \ell_R}{\partial \kappa_i \partial \kappa_j} &= \frac{1}{2} \text{tr}(\mathbf{P}\mathbf{H}_{ij}) - \frac{1}{2} \text{tr}(\mathbf{P}\mathbf{H}_i\mathbf{P}\mathbf{H}_j) \\ &\quad + \mathbf{y}'\mathbf{P}\mathbf{H}_i\mathbf{P}\mathbf{H}_j\mathbf{P}\mathbf{y} - \frac{1}{2} \mathbf{y}'\mathbf{P}\mathbf{H}_{ij}\mathbf{P}\mathbf{y} \end{aligned} \quad (2.7)$$

where $\mathbf{H}_{ij} = \partial^2 \mathbf{H} / \partial \kappa_i \partial \kappa_j$.

The elements of the expected information matrix are

$$\mathbb{E} \left(-\frac{\partial^2 \ell_R}{\partial \kappa_i \partial \kappa_j} \right) = \frac{1}{2} \text{tr}(\mathbf{P}\mathbf{H}_i\mathbf{P}\mathbf{H}_j). \quad (2.8)$$

Given an initial estimate $\boldsymbol{\kappa}^{(0)}$, an update of $\boldsymbol{\kappa}$, $\boldsymbol{\kappa}^{(1)}$ using the Fisher-scoring (FS) algorithm is

$$\boldsymbol{\kappa}^{(1)} = \boldsymbol{\kappa}^{(0)} + \mathbf{I}(\boldsymbol{\kappa}^{(0)}, \boldsymbol{\kappa}^{(0)})^{-1} \mathbf{U}(\boldsymbol{\kappa}^{(0)}) \quad (2.9)$$

where $\mathbf{U}(\boldsymbol{\kappa}^{(0)})$ is the score vector (2.6) and $\mathbf{I}(\boldsymbol{\kappa}^{(0)}, \boldsymbol{\kappa}^{(0)})$ is the expected information matrix (2.8) of $\boldsymbol{\kappa}$ evaluated at $\boldsymbol{\kappa}^{(0)}$.

For large models or large data sets, the evaluation of the trace terms in either (2.7) or (2.8) is either not feasible or is very computer intensive. To overcome this problem ASReml uses the AI algorithm (Gilmour, Thompson and Cullis, 1995). The matrix denoted by \mathcal{I}_A is obtained by averaging (2.7) and (2.8) and approximating $\mathbf{y}'\mathbf{P}\mathbf{H}_{ij}\mathbf{P}\mathbf{y}$ by its expectation, $\text{tr}(\mathbf{P}\mathbf{H}_{ij})$ in those cases when $\mathbf{H}_{ij} \neq 0$. For variance components models (that is those linear with respect to variances in \mathbf{H}), the terms in \mathcal{I}_A are exact averages of those in (2.7) and (2.8). The basic idea is to use $\mathcal{I}_A(\kappa_i, \kappa_j)$ in place of the expected information matrix in (2.9) to update $\boldsymbol{\kappa}$.

The elements of \mathcal{I}_A are

$$\mathcal{I}_A(\kappa_i, \kappa_j) = \frac{1}{2} \mathbf{y}'\mathbf{P}\mathbf{H}_i\mathbf{P}\mathbf{H}_j\mathbf{P}\mathbf{y}. \quad (2.10)$$

The \mathcal{I}_A matrix is the (scaled) residual sums of squares and products matrix of

$$\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_k]$$

where \mathbf{y}_i is the ‘working’ variate for κ_i and is given by

$$\begin{aligned} \mathbf{y}_i &= \mathbf{H}_i \mathbf{P} \mathbf{y} \\ &= \mathbf{H}_i \mathbf{R}^{-1} \tilde{\mathbf{e}} \\ &= \mathbf{R}_i \mathbf{R}^{-1} \tilde{\mathbf{e}}, \quad \kappa_i \in \phi \\ &= \mathbf{Z} \mathbf{G}_i \mathbf{G}^{-1} \tilde{\mathbf{u}}, \quad \kappa_i \in \gamma \end{aligned}$$

where $\tilde{\mathbf{e}} = \mathbf{y} - \mathbf{X} \hat{\boldsymbol{\tau}} - \mathbf{Z} \tilde{\mathbf{u}}$, $\hat{\boldsymbol{\tau}}$ and $\tilde{\mathbf{u}}$ are solutions to (2.11). In this form the AI matrix is relatively straightforward to calculate.

The combination of the AI algorithm with sparse matrix methods, in which only non-zero values are stored, gives an efficient algorithm in terms of both computing time and workspace.

Estimation/prediction of the fixed and random effects

To estimate $\boldsymbol{\tau}$ and predict \mathbf{u} the objective function

$$\log f_{\mathbf{Y}}(\mathbf{y} \mid \mathbf{u}; \boldsymbol{\tau}, \mathbf{R}) + \log f_{\mathbf{U}}(\mathbf{u}; \mathbf{G})$$

is used. This is the log-joint distribution of (\mathbf{Y}, \mathbf{u}) .

Differentiating with respect to $\boldsymbol{\tau}$ and \mathbf{u} leads to the mixed model equations (Robinson, 1991) which are given by

$$\begin{bmatrix} \mathbf{X}' \mathbf{R}^{-1} \mathbf{X} & \mathbf{X}' \mathbf{R}^{-1} \mathbf{Z} \\ \mathbf{Z}' \mathbf{R}^{-1} \mathbf{X} & \mathbf{Z}' \mathbf{R}^{-1} \mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\tau}} \\ \tilde{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}' \mathbf{R}^{-1} \mathbf{y} \\ \mathbf{Z}' \mathbf{R}^{-1} \mathbf{y} \end{bmatrix}. \quad (2.11)$$

These can be written as

$$\mathbf{C} \tilde{\boldsymbol{\beta}} = \mathbf{W} \mathbf{R}^{-1} \mathbf{y}$$

where $\mathbf{C} = \mathbf{W}' \mathbf{R}^{-1} \mathbf{W} + \mathbf{G}^*$, $\tilde{\boldsymbol{\beta}} = [\boldsymbol{\tau}' \quad \mathbf{u}']'$ and

$$\mathbf{G}^* = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} \end{bmatrix}.$$

The solution of (2.11) requires values for γ and ϕ . In practice we replace γ and ϕ by their REML estimates $\hat{\gamma}$ and $\hat{\phi}$.

Note that $\hat{\tau}$ is the best linear unbiased estimator (BLUE) of τ , while $\tilde{\mathbf{u}}$ is the best linear unbiased predictor (BLUP) of \mathbf{u} for known γ and ϕ . We also note that

$$\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta} = \begin{bmatrix} \hat{\tau} - \tau \\ \tilde{\mathbf{u}} - \mathbf{u} \end{bmatrix} \sim N \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \mathbf{C}^{-1} \right).$$

2.3 What are BLUPs?

Consider a balanced one-way classification. For data records ordered r repeats within b treatments, the linear mixed model is $\mathbf{y} = \mathbf{X}\boldsymbol{\tau} + \mathbf{Z}\mathbf{u} + \mathbf{e}$ where $\mathbf{X} = \mathbf{1}_b \otimes \mathbf{1}_r$ is the design matrix for $\boldsymbol{\tau}$, $\mathbf{Z} = \mathbf{I}_b \otimes \mathbf{1}_r$ is the design matrix for the b treatment effects u_i and \mathbf{e} is the error vector. In the following we assume, that the treatment effects are random. That is, $\mathbf{u} \sim N(\mathbf{A}\boldsymbol{\psi}, \sigma_b^2 \mathbf{I}_b)$, for some design matrix \mathbf{A} and parameter vector $\boldsymbol{\psi}$. It can be shown that

$$\tilde{\mathbf{u}} = \frac{b\sigma_b^2}{b\sigma_b^2 + \sigma^2} (\bar{\mathbf{y}} - \mathbf{1}\bar{y}_{..}) + \frac{\sigma^2}{b\sigma_b^2 + \sigma^2} \mathbf{A}\boldsymbol{\psi} \quad (2.12)$$

where $\bar{\mathbf{y}}$ is the vector of treatment means, $\bar{y}_{..}$ is the grand mean. The differences of the treatment means and the grand mean are the estimates of treatment effects if treatment effects are fixed. The BLUP is therefore a weighted mean of the data based estimate and the ‘prior’ mean $\mathbf{A}\boldsymbol{\psi}$. If $\boldsymbol{\psi} = \mathbf{0}$, the BLUP in (2.12) becomes

$$\tilde{\mathbf{u}} = \frac{b\sigma_b^2}{b\sigma_b^2 + \sigma^2} (\bar{\mathbf{y}} - \mathbf{1}\bar{y}_{..}) \quad (2.13)$$

and the BLUP is a so-called shrinkage estimate. As σ_b^2 becomes large relative to σ^2 , the BLUP tends to the fixed effect solution, while for small σ_b^2 relative to σ^2 the BLUP tends towards zero, the assumed initial mean. Thus (2.13) represents a weighted mean which involves the prior assumption that the u_i have zero mean.

Note also that the BLUPs in this simple case are constrained to sum to zero. This is essentially because the unit vector defining \mathbf{X} can be found by summing the columns of the \mathbf{Z} matrix. This linear dependence of the matrices translates to dependence of the BLUPs and hence constraints. This aspect occurs whenever the column space of \mathbf{X} is contained in the column space of \mathbf{Z} . The dependence is slightly more complex with correlated random effects.

2.4 Combining variance models

The combination of variance models within G structures and R structures and between G structures and R structures is a difficult and important concept. The underlying principle is that each R_i and G_i variance model can only have a single scaling variance parameter associated with it. If there is more than one scaling variance parameter for any R_i or G_i then this results in the variance model being overspecified, or *nonidentifiable*. Some variance models are presented in Table 2.1 to illustrate this principle.

While all 9 forms of model in Table 2.1 can be specified within ASReml only models of forms 1 and 2 are recommended. Models 4-6 have too few variance parameters and are likely to cause serious estimation problems. For model 3, where the scale parameter θ has been fitted (univariate single site analysis), it becomes the scale for G . This parameterisation is bizarre and is not recommended. Models 7-9 have too many variance parameters and ASReml will arbitrarily fix one of the variance parameters leading to possible confusion for the user. If you fix the variance parameter to a particular value then it does not count for the purposes of applying the principle that there be only one scaling variance parameter. That is, models 7-9 can be made identifiable by fixing all but one of the nonidentifiable scaling parameters in each of G and R to a particular value.

Table 2.1 Combination of models for G and R structures

model	G_1	G_2	R_1	R_2	θ	comment
1.	V	C	C	C	y	valid
2.	V	C	V	C	n	valid
3.	C	C	V	C	y	valid, but not recommended
4.	*	*	C	C	n	inappropriate as R is a correlation model
5.	C	C	C	C	y	inappropriate, same scale for R and G
6.	C	C	V	C	n	inappropriate, no scaling parameter for G
7.	V	V	*	*	*	nonidentifiable, 2 scaling parameters for G
8.	V	C	V	C	y	nonidentifiable, scale for R and overall scale
9.	*	*	V	V	*	nonidentifiable, 2 scaling parameters for R

* indicates the entry is not relevant in this case

Note that G_1 and G_2 are interchangeable in this table, as are R_1 and R_2

2.5 Inference

Tests of hypotheses: variance parameters

Inference concerning variance parameters of a linear mixed effects model usually relies on approximate distributions for the (RE)ML estimates derived from asymptotic results.

It can be shown that the approximate variance matrix for the REML estimates is given by the inverse of the expected information matrix (Cox and Hinkley, 1974, section 4.8). Since this matrix is not available in ASReml we replace the expected information matrix by the AI matrix. Furthermore the REML estimates are consistent and asymptotically normal, though in small samples this approximation appears to be unreliable (see later).

A general method for comparing the fit of nested models fitted by REML is the REML likelihood ratio test, or REMLRT. The REMLRT is only valid if the fixed effects are the same for both models. In ASReml this requires not only the same fixed effects model, but also the same parameterisation.

If ℓ_{R2} is the REML log-likelihood of the more general model and ℓ_{R1} is the REML log-likelihood of the restricted model (that is, the REML log-likelihood under the null hypothesis), then the REMLRT is given by

$$D = 2 \log(\ell_{R2}/\ell_{R1}) = 2 [\log(\ell_{R2}) - \log(\ell_{R1})] \quad (2.14)$$

which is strictly positive. If r_i is the number of parameters estimated in model i , then the asymptotic distribution of the REMLRT, under the restricted model is $\chi^2_{r_2-r_1}$.

The REMLRT is implicitly two-sided, and must be adjusted when the test involves an hypothesis with the parameter on the boundary of the parameter space. In fact, theoretically it can be shown that for a single variance component, say, the asymptotic distribution of the REMLRT is a mixture of χ^2 variates, where the mixing probabilities are 0.5, one with 0 degrees of freedom (spike at 0) and the other with 1 degree of freedom. The distribution of the REMLRT for the test that k variance components are zero, or tests involved in random regressions, which involve both variance and covariance components, involves a mixture of χ^2 variates from 0 to k degrees of freedom. See Self and Liang (1987) for details.

Tests concerning variance components in generally balanced designs, such as the balanced one-way classification, can be derived from the usual analysis of variance. It can be shown that the REMLRT for a variance component being zero is

a monotone function of the F-statistic for the associated term.

To compare two (or more) non-nested models we can evaluate the *Akaike Information Criteria* (AIC) or the *Bayesian Information Criteria* (BIC) for each model. These are given by

$$\begin{aligned} \text{AIC} &= -2\ell_{Ri} + 2t_i \\ \text{BIC} &= -2\ell_{Ri} + t_i \log \nu \end{aligned} \quad (2.15)$$

where t_i is the number of variance parameters in model i and $\nu = n - p$ is the residual degrees of freedom. AIC and BIC are calculated for each model and the model with the smallest value is chosen as the preferred model.

Diagnostics

In this section we will briefly review some of the diagnostics that have been implemented in ASReml for examining the adequacy of the assumed variance matrix for either R or G structures, or for examining the distributional assumptions regarding \mathbf{e} or \mathbf{u} . Firstly we note that the BLUP of the residual vector is given by

$$\begin{aligned} \tilde{\mathbf{e}} &= \mathbf{y} - \mathbf{W}\tilde{\boldsymbol{\beta}} \\ &= \mathbf{R}\mathbf{P}\mathbf{y} \end{aligned} \quad (2.16)$$

It follows that

$$\begin{aligned} \mathbf{E}(\tilde{\mathbf{e}}) &= \mathbf{0} \\ \text{var}(\tilde{\mathbf{e}}) &= \mathbf{R} - \mathbf{W}\mathbf{C}^{-1}\mathbf{W}' \end{aligned}$$

The matrix $\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'$ is the so-called ‘extended hat’ matrix. It is the linear mixed effects model analogue of $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ for ordinary linear models. The diagonal elements are returned in the `.yht` file by ASReml.

The variogram has been suggested as a useful diagnostic for assisting with the identification of appropriate variance models for spatial data (Cressie, 1991). Gilmour *et al.* (1997) demonstrate its usefulness for the identification of the sources of variation in the analysis of field experiments. If the elements of the data vector (and hence the residual vector) are indexed by a vector of spatial coordinates, $\mathbf{s}_i, i = 1, \dots, n$, then the ordinates of the sample variogram are given by

$$v_{ij} = \frac{1}{2} [\tilde{e}_i(\mathbf{s}_i) - \tilde{e}_j(\mathbf{s}_j)], \quad i, j = 1, \dots, n; \quad i \neq j$$

The sample variogram is the triple $(l_{ij1}, l_{ij2}, v_{ij})$ where $l_{ij1} = |s_{i1} - s_{j1}|$ and $l_{ij2} = |s_{i2} - s_{j2}|$ are the absolute displacements. If the data arise from a regular

array there will be many v_{ij} with the same absolute displacements, in which case ASReml displays the vector $(l_{ij1}, l_{ij2}, \bar{v}_{ij})$ as a perspective plot and returns it in the .res file.

Tests of hypotheses: fixed effects

Testing the significance of fixed effects under REML does introduce a difficulty. The standard approach is to use likelihood ratio methods. Unfortunately, if the fixed effects part of the model is changed, the residual likelihood changes. Thus the maximized residual likelihood under the null and alternative hypotheses are not comparable. Alternative methods are required.

An obvious approach is to use the Wald statistic. To test the hypothesis $H_0 : \mathbf{L}\boldsymbol{\tau} = \mathbf{l}$ for given \mathbf{L} , $r \times p$, and \mathbf{l} , $r \times 1$, the Wald statistic is given by

$$W = (\mathbf{L}\hat{\boldsymbol{\tau}} - \mathbf{l})' \{ \mathbf{L}(\mathbf{X}'\mathbf{H}^{-1}\mathbf{X})^{-1}\mathbf{L}' \}^{-1} (\mathbf{L}\hat{\boldsymbol{\tau}} - \mathbf{l}) \quad (2.17)$$

and asymptotically, this statistic has a chi-square distribution on r degrees of freedom. These are marginal tests, so that there is an adjustment for all other terms in the fixed part of the model.

It is also possible to carry out a sequential test so that higher order terms, for example interactions, are adjusted for lower order terms in the model. The latter tests conform more to the analysis of variance philosophy. The test statistic implicit in this sequential approach is the change in residual sum of squares between the model not containing the term and that containing the term.

ASReml produces an F-statistic by dividing the Wald test by r . In this form it is possible to perform an approximate F test. The numerator degrees of freedom for this test is r . The denominator degrees of freedom is generally not known exactly. It is possible, though anti-conservative for most situations to use the residual degrees of freedom for the denominator. For balanced designs the F statistics produced by ASReml are numerically identical to the F-tests obtained from the standard analysis of variance.

3

A guided tour

Introduction

Nebraska Intrastate Nursery (NIN) field experiment

The ASReml data file

The ASReml command file

- The title line
- Reading the data
- The data file line
- Specifying the terms in the mixed model
- Tabulation
- Prediction
- Variance structures

Running the job

Description of output files

- The .asr file
- The .sln file
- The .yht file

Tabulation, predicted values and functions of the variance components

3.1 Introduction

This chapter presents a guided tour of ASReml, from data file preparation and basic aspects of the ASReml command file, to running an ASReml job and interpreting the output files. You are encouraged to read this chapter before moving to the later chapters;

- a real data example is used in this chapter for demonstration, see below,
- the same data are also used in later chapters,
- links to the formal discussion of topics are clearly signposted by margin notes.

Note that some aspects of ASReml, in particular, pedigree files (see Chapter 9) and multivariate analysis (see Chapter 8) are only covered in later chapters.

ASReml is usually run in batch mode from a command prompt. You will need a file editor to create the command file and to view the various output files. On unix systems, `vi` and `emacs` are commonly used. Under Windows, there are several program editors available such as `Brief` and `Winedt`. You can use `NOTEPAD` or `WORDPAD`. ASReml has a built in editor which is accessed through `menu` mode (page).

3.2 Nebraska Intrastate Nursery (NIN) field experiment

The yield data from an advanced Nebraska Intrastate Nursery (NIN) breeding trial conducted at Alliance in 1988/89 will be used for demonstration, see Stroup *et al.* (1994) for details. Four replicates of 19 released cultivars, 35 experimental wheat lines and 2 additional triticale lines were laid out in a 22 row by 11 column rectangular array of plots; the varieties were allocated to the plots using a randomised complete block (RCB) design. In field trials, complete replicates are typically allocated to consecutive groups of *whole* columns or rows. In this trial the replicates were not allocated to groups of whole columns, but rather, overlapped columns. Table 3.1 gives the allocation of varieties to plots in field plan order with replicates 1 and 3 in *ITALICS* and replicates 2 and 4 in **BOLD**.

3.1: Trial layout and allocation of varieties to plots in the N11N field trial

	1	2	3	4	5	6	7	8	9	10	11
-		NE83407	BUCKSKIN	NE87612	VONA	NE87512	NE87408	CODY	BUCKSKIN	NE87612	KS831374
-		CENTURA	NE86527	NE87613	NE87463	NE83407	NE83407	NE87612	NE83406	BUCKSKIN	NE86482
-		SCOUT66	NE86582	NE87615	NE86507	NE87403	NORKAN	NE87457	NE87409	NE85556	NE85623
-		COLT	NE86606	NE87619	BUCKSKIN	NE87457	REDLAND	NE84557	NE87499	BRULE	NE86527
-		NE83498	NE86607	NE87627	ROUGHHRIDER	NE83406	KS831374	NE83T12	CENTURA	NE86507	NE87451
-		NE84557	ROUGHHRIDER	-	NE86527	COLT	COLT	NE86507	NE83432	ROUGHHRIDER	NE87409
-		NE83432	VONA	CENTURA	SCOUT66	NE87522	NE86527	TAM200	NE87512	VONA	GAGE
-		NE85556	SIUXLAND	NE85623	NE86509	NORKAN	VONA	NE87613	ROUGHHRIDER	NE83404	NE83407
-		NE85623	GAGE	CODY	NE86606	NE87615	TAM107	ARAPAHOE	NE83498	CODY	NE87615
-		CENTURAK78	NE83T12	NE86582	NE84557	NE85556	CENTURAK78	SCOUT66	-	NE87463	ARAPAHOE
-		NORKAN	NE86T666	NE87408	KS831374	TAM200	NE87627	NE87403	NE86T666	NE86582	CHEYENNE
-		KS831374	NE87403	NE87451	GAGE	LANCOTA	NE86T666	NE85623	NE87403	NE87499	REDLAND
-		TAM200	NE87408	NE83432	NE87619	NE86503	NE87615	NE86509	NE87512	NORKAN	NE83432
-		NE86482	NE87409	CENTURAK78	NE87499	NE86482	NE86501	NE85556	NE87446	SCOUT66	NE87619
-		HOMESTEAD	NE87446	NE83T12	CHEYENNE	BRULE	NE87522	HOMESTEAD	CENTURA	NE87513	NE83498
LANCER		LANCOTA	NE87451	NE87409	NE86607	NE87612	CHEYENNE	NE83404	NE86503	NE83T12	NE87613
BRULE		NE86501	NE87457	NE87513	NE83498	NE87613	SIUXLAND	NE86503	NE87408	CENTURAK78	NE86501
REDLAND		NE86503	NE87463	NE87627	NE83404	NE86T666	NE87451	NE86582	COLT	NE87627	TAM200
CODY		NE86507	NE87499	ARAPAHOE	NE87446	-	GAGE	NE87619	LANCER	NE86606	NE87522
ARAPAHOE		NE86509	NE87512	LANCER	SIUXLAND	NE86607	LANCER	NE87463	NE83406	NE87457	NE84557
NE83404		TAM107	NE87513	TAM107	HOMESTEAD	LANCOTA	NE87446	NE86606	NE86607	NE86509	TAM107
NE83406		CHEYENNE	NE87522	REDLAND	NE86501	NE87513	NE86482	BRULE	SIUXLAND	LANCOTA	HOMESTEAD

3.3 The ASReml data file

See Chapter 4 for details

The standard format of an ASReml data file is to have the data arranged in space, TAB or comma separated columns/fields with a line for each sampling unit. The columns contain covariates, factors, response variates (traits) and weight variables in any convenient order. This is the first 30 lines of the file `nin89.asd` containing the data for the NIN variety trial. The data are in field order (rows within columns) and an optional row of field labels (first row) has been included. In this case there are 11 space separated data fields (`variety...column`) and the complete file has 224 data rows, one for each variety in each replicate.

```
variety id pid raw repl nloc yield lat long row column
LANCER 1 1101 585 1 4 29.25 4.3 19.2 16 1
BRULE 2 1102 631 1 4 31.55 4.3 20.4 17 1
REDLAND 3 1103 701 1 4 35.05 4.3 21.6 18 1
CODY 4 1104 602 1 4 30.1 4.3 22.8 19 1
ARAPAHOE 5 1105 661 1 4 33.05 4.3 24 20 1
NE83404 6 1106 605 1 4 30.25 4.3 25.2 21 1
NE83406 7 1107 704 1 4 35.2 4.3 26.4 22 1
NE83407 8 1108 388 1 4 19.4 8.6 1.2 1 2
CENTURA 9 1109 487 1 4 24.35 8.6 2.4 2 2
SCOUT66 10 1110 511 1 4 25.55 8.6 3.6 3 2
COLT 11 1111 502 1 4 25.1 8.6 4.8 4 2
NE83498 12 1112 492 1 4 24.6 8.6 6 5 2
NE84557 13 1113 509 1 4 25.45 8.6 7.2 6 2
NE83432 14 1114 268 1 4 13.4 8.6 8.4 7 2
NE85556 15 1115 633 1 4 31.65 8.6 9.6 8 2
NE85623 16 1116 513 1 4 25.65 8.6 10.8 9 2
CENTURAK78 17 1117 632 1 4 31.6 8.6 12 10 2
NORKAN 18 1118 446 1 4 22.3 8.6 13.2 11 2
KS831374 19 1119 684 1 4 34.2 8.6 14.4 12 2
TAM200 20 1120 422 1 4 21.1 8.6 15.6 13 2
NE86482 21 1121 560 1 4 28 8.6 16.8 14 2
HOMESTEAD 22 1122 566 1 4 28.3 8.6 18 15 2
LANCOTA 23 1123 514 1 4 25.7 8.6 19.2 16 2
NE86501 24 1124 635 1 4 31.75 8.6 20.4 17 2
NE86503 25 1125 840 1 4 42 8.6 21.6 18 2
NE86507 26 1126 618 1 4 30.9 8.6 22.8 19 2
NE86509 27 1127 658 1 4 32.9 8.6 24 20 2
TAM107 28 1128 481 1 4 24.05 8.6 25.2 21 2
CHEYENNE 29 1129 564 1 4 28.2 8.6 26.4 22 2
:
```

optional field labels
data line sampling unit 1
data line sampling unit 2

Note that in Chapter 7 these data are analysed using spatial methods of analysis, see model **3a** in Section 7.3. For spatial analysis using a separable error structure (see Chapter 2) the data file must first be augmented to specify the complete 22

row \times 11 column array of plots. These are the first 20 lines of the augmented data file `nin89aug.asd` with 242 data rows.

```

variety id pid raw repl nloc yield lat long row column
LANCER 1 NA NA 1 4 NA 4.3 1.2 1 1
LANCER 1 NA NA 1 4 NA 4.3 2.4 2 1
LANCER 1 NA NA 1 4 NA 4.3 3.6 3 1
LANCER 1 NA NA 1 4 NA 4.3 4.8 4 1
LANCER 1 NA NA 1 4 NA 4.3 6 5 1
LANCER 1 NA NA 1 4 NA 4.3 7.2 6 1
LANCER 1 NA NA 1 4 NA 4.3 8.4 7 1
LANCER 1 NA NA 1 4 NA 4.3 9.6 8 1
LANCER 1 NA NA 1 4 NA 4.3 10.8 9 1
LANCER 1 NA NA 1 4 NA 4.3 12 10 1
LANCER 1 NA NA 1 4 NA 4.3 13.2 11 1
LANCER 1 NA NA 1 4 NA 4.3 14.4 12 1
LANCER 1 NA NA 1 4 NA 4.3 15.6 13 1
LANCER 1 NA NA 1 4 NA 4.3 16.8 14 1
LANCER 1 NA NA 1 4 NA 4.3 18 15 1
LANCER 1 NA NA 2 4 NA 17.2 7.2 6 4
LANCER 1 NA NA 3 4 NA 25.8 22.8 19 6
LANCER 1 NA NA 4 4 NA 38.7 12.0 10 9
LANCER 1 1101 585 1 4 29.25 4.3 19.2 16 1
BRULE 2 1102 631 1 4 31.55 4.3 20.4 17 1
REDLAND 3 1103 701 1 4 35.05 4.3 21.6 18 1
CODY 4 1104 602 1 4 30.1 4.3 22.8 19 1
:
:
:

```

optional field labels
file augmented by
missing values for first
15 plots and 3 buffer
plots and variety coded
LANCER to complete
22 \times 11 array

.
.

.

buffer plots
between reps

original data

.
.

.

Note that

- the `pid`, `raw`, `repl` and `yield` data for the missing plots have all been made NA (one of the three missing value indicators in ASReml, see Section 4.2),
- `variety` is coded LANCER for all missing plots; one of the variety names must be used but the particular choice is arbitrary.

3.4 The ASReml command file

See Chapters 5, 6 and 7 for details. By convention an ASReml command file has a `.as` extension. The file defines

- a title line to describe the job,
- labels for the data fields in the data file and the name of the data file,
- the linear mixed model and the variance model(s) if required,
- output options including directives for tabulation and prediction.

Below is the ASReml command file for an RCB analysis of the NIN field trial data highlighting the main sections. Notice the order of the main sections.

title line →	NIN Alliance trial 1989
data field definition →	variety !A
	id
	pid
	raw
	repl 4
	nloc
	yield
	lat
	long
	row 22
data field definition →	column 11
data file name and qualifiers →	nin89.asd !skip 1
linear mixed model definition →	yield ~ mu variety !r repl
tabulate statement →	tabulate yield ~ variety
predict statement →	predict variety
variance model specification →	0 0 1
	repl 1
	repl 0 IDV 0.1

The title line

The first non-blank line in an ASReml command file is taken as the title for the job and is used to identify the analysis for future reference.

```
NIN Alliance trial 1989
variety !A
id
:
:
```

Reading the data

The data fields are defined prior to the data file name to tell ASReml how many fields to expect in the data file and what the fields are. Field definitions must be given for all fields in the data file and in the order in which they appear in the data file. **Data field definitions must be indented.** In this case there are 11 data fields in the data file `nin89.asd`, see Section 3.3. They are specified in the command file in the order they appear in the data file, that is, `variety ... column`. The `!A` after `variety` tells ASReml that the first field is al-

```
NIN Alliance trial 1989
variety !A
id
pid
raw
repl 4
nloc
yield
lat
long
row 22
column 11
nin89.asd !skip 1
:
:
```

phanumeric and the 4 after `repl` tells ASReml that the field called `repl` (the fifth field read) is a numeric factor with 4 levels. Similarly for `row` and `column`. The other data fields include variates (`yield`) and various other variables.

The data file line

The data file name is specified immediately after the last data field definition. Data file qualifiers that relate to data input and output are also placed on this line if they are required.

See Section 5.7

In this example, `!skip 1` tells ASReml to read the data from `nin89.asd` but to ignore the first line in this file, the line containing the field labels.

See Section 5.8

The data file line can also contain qualifiers that control other aspects of the analysis. These qualifiers are presented in Section 5.8.

```
NIN Alliance trial 1989
variety !A
id
pid
:
:
row 22
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
tabulate yield ~ variety
predict variety
0 0 1
repl 1
repl 0 IDV 0.1
```

Specifying the terms in the mixed model

See Chapter 6

The linear mixed model is specified as a list of model terms and qualifiers. All elements must be space separated. ASReml accommodates a wide range of analyses. See Section 2.1 for a brief discussion and general algebraic formulation of the linear mixed model. The model specified here for the NIN data is a simple random effects RCB model including fixed variety effects and random replicate effects. The reserved word `mu` fits a constant term (intercept), `variety` fits a fixed variety effect and `repl` fits a random replicate effect. The `!r` qualifier tells ASReml to fit the terms that appear after this qualifier as random effects.

```
NIN Alliance trial 1989
variety !A
:
:
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
tabulate yield ~ variety
predict variety
0 0 1
repl 1
repl 0 IDV 0.1
```

Tabulation

See Chapter 10 Tabulate statements provide a simple way of exploring the structure of a data file. They appear after the model line and before any variance structure definition lines. In this case the 56 simple variety means for yield are formed and written to a `.tab` output file. See Chapter 10 for a discussion of tabulation.

```
NIN Alliance trial 1989
  variety !A
  :
  column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
tabulate yield ~ variety
predict variety
0 0 1
repl 1
repl 0 IDV 0.1
```

Prediction

See Chapter 10 Prediction statements also appear after the model statement and before any variance structure lines. In this case the 56 variety means for yield would be formed and returned in the `.pvs` output file. See Chapter 10 for a detailed discussion of prediction in ASReml.

```
NIN Alliance trial 1989
  variety !A
  :
  column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
tabulate yield ~ variety
predict variety
0 0 1
repl 1
repl 0 IDV 0.1
```

Variance structures

See Chapter 7 An extensive range of variance structures can be fitted in ASReml. See Chapter 7 for a lengthy discussion of variance modelling in ASReml. In this case independent and identically distributed random replicate effects are specified using the identifier IDV in a *G structure*. G structures are described in Section 2.1 and the list of available variance structures/models is presented in Table 7.3. Note that IDV is the default variance structure for random effects so that the last 3 lines presented here are purely for example: the same analysis would be performed if these lines were omitted.

```
NIN Alliance trial 1989
  variety !A
  :
  column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
tabulate yield ~ variety
predict variety
0 0 1
repl 1
repl 0 IDV 0.1
```

3.5 Running the job

See Chapter 12 An ASReml job is often run from a command line. The basic command to run an ASReml job is normally

```
asreml basename [.as]
```

where *basename* [.as] is the name of the command file. For example, the command to run `nin89.as` is

```
asreml nin89.as
```

However, if the path to program `asreml.exe` is not specified in your system's PATH environment variable, a *path* to this program must also be given. In this case the command to run the ASReml job in a PC/Windows environment is

```
[path] asreml basename [.as]
```

where *path* might be `c:\asreml\` pointing to the location of the ASReml program.

Make a habit of giving command files the .as extension ASReml is always run using an ASCII text command file. In this guide we assume the command file has a filename extension `.as`. ASReml also recognises the filename extension `.asc` as an ASReml command file. When these are used, the extension (`.as` or `.asc`) may be omitted from *basename.as* in the command line if there is no file in the same sub-directory with the name *basename*. There are *options* and *arguments* that can be supplied on the command line to modify a job at run time. These are described in Chapter 12.

3.6 Description of output files

A series of output files are produced with each ASReml run. Running the example as above, the primary output is written to `nin89.asr`. This file contains a summary of the data, the iteration sequence, estimates of the variance parameters, an analysis of variance (ANOVA) table and estimates of the fixed effects. The estimates of all the fixed and random effects are written to `nin89.sln`. The residuals, predicted values of the observations and the diagonal elements of the hat matrix (see Chapter 2) are returned in `nin89.yht`, see Section 13.3. Other files include the `.res`, `.asl`, `.asp`, `.pin`, `.pvc`, `.pvs`, `.tab`, `.vvp` and `.vrb` files, see Section 13.4.

The .asr file

Below is `nin89.asr` with pointers to the main sections. The title line gives the version of ASReml used (in square brackets) and the title of the job. The second

line gives the date and time that the job was run. The following output presents a data summary, the iteration sequence, the estimated variance parameters, an ANOVA and estimates of the fixed effects. The second to last line notifies the user of possible outliers as recorded in the `.res` file and the final line gives the date and time that the job was completed and a statement about convergence. The general announcements box (outlined in asterisks) at the top of the file notifies the user of current release features.

```

title line      ASReml 1.0 [31 Aug 2002]  nin alliance trial
date and time  13 Sep 2002 12:55:17.777  8.00 Mbyte  Windows  nin89
release features
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support  *
* Use !{ and !} when fitting covariance across terms  *
***** ARG *
QUALIFIERS: !SKIP 1
data summary   Reading nin89.asd  FREE FORMAT skipping 1 lines
                Folder: C:\asr\ex\manex
                Univariate analysis of yield
                Using      224 records of      224 read
                Model term      Size Type  COL  Minimum  Mean      Maximum  #zero #miss
                1 variety      56 Factor  1    1      28.5000   56        0    0
                2 id           2 1.000   2    1.000   28.50     56.00     0    0
                3 pid          3 1101.   3    1101.   2629.     4156.     0    0
                4 raw          4 21.00   4    21.00   510.5     840.0     0    0
                5 repl         4 Factor  5    1      2.5000    4         0    0
                6 nloc         6 4.000   6    4.000   4.000     4.000     0    0
                7 yield        1 Variate 7    1.050   25.53     42.00     0    0
                8 lat          8 4.300   8    4.300   27.22     47.30     0    0
                9 long         9 1.200   9    1.200   14.08     26.40     0    0
                10 row         22 Factor 10    1     11.7321   22        0    0
                11 column      11 Factor 11    1      6.3304    11        0    0
                12 mu          1 Constant Term
                Forming 61 equations: 57 dense
                Initial updates will be shrunk by factor 0.316
iteration       NOTICE:      1 (more) singularities,
sequence       1 LogL=-454.807  S2= 50.329      168 df  0.1000   1.000
               2 LogL=-454.663  S2= 50.120      168 df  0.1173   1.000
               3 LogL=-454.532  S2= 49.868      168 df  0.1463   1.000
               4 LogL=-454.472  S2= 49.637      168 df  0.1866   1.000
               5 LogL=-454.469  S2= 49.585      168 df  0.1986   1.000
               6 LogL=-454.469  S2= 49.582      168 df  0.1993   1.000
Final parameter values
               0.19932   1.0000

estimated      Source          Model terms      Gamma      Component      Comp/SE  % C
variance      repl             4      4  0.199323      9.88291      1.12  0 P
parameters    Variance         224    168  1.00000      49.5824      9.08  0 P

```



```

ANOVA      Analysis of Variance      DF      F-incr
           1 variety                  55      0.88
NOTICE: ASREML cannot yet determine the correct denominator DFs for
           these F statistics. They will not exceed the Residual DF
           which applies only if all components are KNOWN.

estimated effects
           Estimate      Standard Error      T-value      T-prev
           1 variety
BRULE      -2.48750      4.97908      -0.50
REDLAND    1.93750      4.97908      0.39      0.89
CODY       -7.35000      4.97908      -1.48     -1.87
ARAPAHOE   0.875000      4.97908      0.18      1.65
NE83404    -1.17500      4.97908      -0.24     -0.41
           ⋮
NE87619    2.70000      4.97908      0.54      1.12
NE87627    -5.33750      4.97908      -1.07     -1.61
           12 mu
           57      28.5625      3.85568      7.41

           5 repl      4 effects fitted

outlier info      1 possible outliers: see .res file
completion info    Finished: 13 Sep 2002 12:55:22.133      LogL Converged

```

The .sln file

The following is an extract from `nin89.sln` containing the estimated variety effects, intercept and random replicate effects in this order (column 3) with standard errors (column 4). Note that the variety effects are returned in the order of their first appearance in the data file, see replicate 1 in Table 3.1. Note also that LANCER did not appear in the list in the `.asr` file because it was aliased with `mu` (see 6.9) but it does appear here.

```

variety      LANCER      0.000      0.000
variety      BRULE      -2.487      4.979
variety      REDLAND    1.938      4.979
variety      CODY       -7.350      4.979
variety      ARAPAHOE   0.87500    4.979
variety      NE83404    -1.175     4.979
variety      NE83406    -4.287     4.979
variety      NE83407    -5.875     4.979

```

variety	CENTURA		-6.912	4.979
variety	SCOUT66		-1.037	4.979
variety	COLT		-1.562	4.979
variety	NE83498		1.563	4.979
variety	NE84557		-8.037	4.979
variety	NE83432		-8.837	4.979
:				
variety	NE87615		-2.875	4.979
variety	NE87619		2.700	4.979
variety	NE87627		-5.337	4.979
mu		1	28.56	3.856
repl		1	1.880	1.755
repl		2	2.843	1.755
repl		3	-0.8713	1.755
repl		4	-3.852	1.755

The .yht file

The following is an extract from `nin89.yht` containing the predicted values of the observations (column 2), the residuals (column 3) and the diagonal elements of the hat matrix. This final column can be used in tests involving the residuals, see Section 2.5 under Diagnostics.

Record	Yhat	Residual	Hat
1	30.442	-1.192	13.01
2	27.955	3.595	13.01
3	32.380	2.670	13.01
4	23.092	7.008	13.01
5	31.317	1.733	13.01
6	29.267	0.9829	13.01
7	26.155	9.045	13.01
8	24.567	-5.167	13.01
9	23.530	0.8204	13.01
:			
222	16.673	9.877	13.01
223	24.548	1.052	13.01
224	23.786	3.114	13.01

3.7 Tabulation, predicted values and functions of the variance components

It may take several runs of ASReml to determine an appropriate model for the data, that is, the fixed and random effects that are important. During this process you may wish to explore the data by simple tabulation. Having identified an appropriate model, you may then wish to form predicted values or functions of the variance components. The facilities available in ASReml to determine predicted values and functions of the variance components are described in Chapters 10 and 11 respectively. Our example only includes tabulation and prediction.

The statement

```
tabulate yield ~ variety
```

in `nin89.as` results in `nin89.tab` as follows:

```
nin alliance trial
```

```
13 Sep 2002 12:35:40
```

```
Simple tabulation of yield
```

```
variety
LANCER          28.56
BRULE           26.07
REDLAND         30.50
CODY            21.21
ARAPAHOE        29.44
NE83404         27.39
NE83406         24.28
NE83407         22.69
CENTURA        21.65
SCOUT66         27.52
COLT            27.00
:
NE87522         25.00
NE87612         21.80
NE87613         29.40
NE87615         25.69
NE87619         31.26
NE87627         23.23
```

The

`predict variety`

statement after the model statement in `nin89.as` results in the `nin89.pvs` file displayed below (some output omitted) containing the 56 predicted variety means, also in the order in which they first appear in the data file (column 2), together with standard errors (column 3). An average standard error of difference among the predicted variety means is displayed immediately after the list of predicted values. As in the `.asr` file, date, time and trial information are given the title line. The `Ecode` for each prediction (column 4) is usually `E` indicating the prediction is of an estimable function. Predictions of non-estimable functions are usually not printed, see Chapter 10.

```

title line      nin alliance trial                               13 Sep 2002 12:35:40
                                                         nin893

      Ecode is E for Estimable, * for Not Estimable

----- 1 -----
Predicted values of yield
repl          is ignored in the prediction (except where specifically included

variety      Predicted_Value Standard_Error Ecode
predicted variety effects
LANCER              28.5625         3.8557 E
BRULE              26.0750         3.8557 E
REDLAND           30.5000         3.8557 E
CODY              21.2125         3.8557 E
ARAPAHOE         29.4375         3.8557 E
NE83404          27.3875         3.8557 E
NE83406          24.2750         3.8557 E
NE83407          22.6875         3.8557 E
CENTURA         21.6500         3.8557 E
SCOUT66         27.5250         3.8557 E
COLT            27.0000         3.8557 E
:
NE87613          29.4000         3.8557 E
NE87615          25.6875         3.8557 E
NE87619          31.2625         3.8557 E
NE87627          23.2250         3.8557 E
SED: Overall Standard Error of Difference  4.979

```

4

Data file preparation

Introduction

The data file

Free format data files
Binary format data files

4.1 Introduction

The first step in an ASReml analysis is to prepare the data file. Data file preparation is described in this chapter using the NIN example of Chapter 3 for demonstration. The first 25 lines of the data file are as follows:

```
variety id pid raw repl nloc yield lat long row column
BRULE 2 1102 631 1 4 31.55 4.3 20.4 17 1
REDLAND 3 1103 701 1 4 35.05 4.3 21.6 18 1
CODY 4 1104 602 1 4 30.1 4.3 22.8 19 1
ARAPAHOE 5 1105 661 1 4 33.05 4.3 24 20 1
NE83404 6 1106 605 1 4 30.25 4.3 25.2 21 1
NE83406 7 1107 704 1 4 35.2 4.3 26.4 22 1
NE83407 8 1108 388 1 4 19.4 8.6 1.2 1 2
CENTURA 9 1109 487 1 4 24.35 8.6 2.4 2 2
SCOUT66 10 1110 511 1 4 25.55 8.6 3.6 3 2
COLT 11 1111 502 1 4 25.1 8.6 4.8 4 2
NE83498 12 1112 492 1 4 24.6 8.6 6 5 2
NE84557 13 1113 509 1 4 25.45 8.6 7.2 6 2
NE83432 14 1114 268 1 4 13.4 8.6 8.4 7 2
NE85556 15 1115 633 1 4 31.65 8.6 9.6 8 2
NE85623 16 1116 513 1 4 25.65 8.6 10.8 9 2
CENTURK78 17 1117 632 1 4 31.6 8.6 12 10 2
NORKAN 18 1118 446 1 4 22.3 8.6 13.2 11 2
KS831374 19 1119 684 1 4 34.2 8.6 14.4 12 2
TAM200 20 1120 422 1 4 21.1 8.6 15.6 13 2
NE86482 21 1121 560 1 4 28 8.6 16.8 14 2
HOMESTEAD 22 1122 566 1 4 28.3 8.6 18 15 2
LANCOTA 23 1123 514 1 4 25.7 8.6 19.2 16 2
NE86501 24 1124 635 1 4 31.75 8.6 20.4 17 2
NE86503 25 1125 840 1 4 42 8.6 21.6 18 2
:
```

4.2 The data file

The standard format of an ASReml data file is to have the data arranged in columns/fields with a single line for each sampling unit. The columns contain variates and covariates (numeric), factors (alphanumeric), traits (response variables) and weight variables in any order that is convenient to the user. The data file may be free format, fixed format or a binary file.

Free format data files

The data are read free format (SPACE, comma or TAB separated) unless the file name has extension `.bin` for real binary, or `.dbl` for double precision binary (see

below). Important points to note are as follows:

- blank lines are ignored,
- column headings, field labels or comments may be present at the top of the file provided that the `!skip` qualifier (Table 5.2) is used to skip over them,
- `NA`, `*` and `.` are accepted as coding for *missing values* in free format data files;
 - missing values can also be coded with a unique value (for example, 0 or -9) and then using the `!M` or `!D` qualifiers to declare them as missing values (see Table 5.1),
- comma delimited files whose file name ends in `.csv` or for which the `!CSV` qualifier is set recognise empty fields as missing values,
 - a line beginning with a comma implies a preceding missing value,
 - consecutive commas imply a missing value,
 - a line ending with a comma implies a trailing missing value,
 - if the filename does not end in `.csv` or the `!CSV` qualifier is not set, commas are treated as white space,
- characters following `#` on a line are ignored so this character may not be used in alphanumeric fields,
- blank spaces, tabs and commas must not be used (embedded) in alphanumeric fields unless the label is enclosed in quotes, for example, the name `Willow Creek` would need to be appear in the data file as `'Willow Creek'` to avoid error,
- the `$` symbol must not be used in the data file,
- alphanumeric fields are truncated to 20 characters in length,
- extra data fields on a line are ignored,
- if there are fewer data items on a line than `ASReml` expects the remainder are taken from the following line(s) except in `.csv` files were they are taken as missing. If you end up with half the number of records you expected, this is probably the reason,
- all lines beginning with `!` followed by a blank are copied to the `.asr` file as comments for the output; their contents are ignored,
- a data file line may not exceed 418 characters; if the data fields will not fit in 418 characters, put some on the next line.

Fixed format data files

The format must be supplied with the `!FORMAT` qualifier which is described in (Table 5.5). However, if all fields are present and are separated, the file can be read free format.

Binary format data files

Conventions for binary files are as follows:

- binary files are read as unformatted Fortran binary in single precision if the filename has a `.bin` or `.BIN` extension,
- fortran binary data files are read in double precision if the filename has a `.dbl` or `.DBL` extension,
- ASReml recognises the value `-1e37` as a missing value in binary files,
- fortran binary in the above means all real (`.bin`) or all double precision (`.dbl`) variables; mixed types, that is, integer and alphabetic binary representation of variables is not allowed in binary files,
- binary files can only be used in conjunction with a pedigree file if the pedigree fields are coded in the binary file so that they correspond with the pedigree file (this can be done using the `!SAVE` option in ASReml to form the binary file, see Table 5.5), or the identifiers are whole numbers less than 9,999,999 and the `!RECODE` qualifier is specified (see Table 5.5).

5 Command file: Reading the data

Introduction

Important rules

Title line

Specifying and reading the data

Data field definition syntax

Transforming the data

Transformation syntax

Other rules and examples

Special note on covariates

Other examples

Data file line

Data line syntax

Data file qualifiers

Job control qualifiers

5.1 Introduction

In the code box to the right is the ASReml command file `nin89.as` for the Nebraska Intrastate Nursery (NIN) field experiment introduced Chapter 3. The lines that are highlighted in bold/blue type relate to reading in the data. In this chapter we use this example to discuss reading in the data in detail.

```
NIN Alliance Trial 1989
variety !A
id
pid
raw
repl 4
nloc
yield
lat
long
row 22
column 11
alliance89.asd !skip 1
yield ~ mu variety
1 2
11 column AR1 .424
22 row AR1 .904
```

5.2 Important rules

In the ASReml command file

- all characters following a `#` symbol on a line are ignored,
- all blank lines are ignored,
- lines beginning with `!` followed by a blank are copied to the `.asr` file as comments for the output,
- a blank is the usual separator; TAB is also a separator,
- maximum line length is 418 characters but it is usually best to limit lines to 80 characters for ease of editing,
- a comma as the last character on the line is used to indicate that the current list is continued on the next line; a comma is not needed when ASReml knows how many values to read,
- reserved words used in specifying the linear model (Table 6.1) are case sensitive; they need to be typed exactly as defined: they may not be abbreviated.
- a qualifier is a particular letter sequence beginning with an `!` which sets an option or changes some aspect of ASReml;
 - some qualifiers require arguments,
 - qualifiers must appear on the correct line,
 - qualifier identifiers are not case sensitive,
 - qualifier identifiers may be truncated to 3 characters.

5.3 Title line

The first 40 characters of the first nonblank line in an ASReml command file are taken as a title for the job. Use this to identify the analysis for future reference.

```
NIN Alliance Trial 1989
variety !A
id
pid
:
:
```

5.4 Specifying and reading the data

The data fields are defined immediately after the job title. They tell ASReml how many fields to expect in the data file and what they are. Data field definitions

- should be given for all fields in the data file; data fields on the end of a data line that do not have a corresponding field definition will be ignored,
- must be presented in the order in which they appear in the data file,
- must be indented one or more spaces,
- can appear with other definitions on the same line,
- data fields can be transformed as they are defined (see below),
- additional data fields can be created by transformation; these should be listed after the data fields read from the data file.

Important

```
NIN Alliance Trial 1989
variety !A
id
pid
raw
repl 4
nloc
yield
lat
long
row 22
column 11
alliance89.asd !skip 1
yield ~ mu variety
:
:
```

Data field definition syntax

Data field definitions appear in the ASReml command file in the form

```
SPACE label [field.type] [transformations]
```

- SPACE
 - is a required space
- *label*
 - is an alphanumeric string to identify the field,

- has a maximum of 31 characters of which only 20 are printed; the remaining characters are not displayed,
 - must begin with a letter,
 - must not contain the special characters `.`, `*`, `:`, `/`, `!`, `#`, `—` or `(`,
 - reserved words (Table 6.1 and Table 7.3) must not be used,
- *field_type* defines how a variable is interpreted if specified in the linear model,
 - for a variate, leave *field_type* blank or specify 1,
 - for a model factor, various qualifiers are required depending on the form of the factor coding where *n* is the number of levels of the factor and *s* is a list of labels to be assigned to the levels:
 - * or *n* is used when the data field has values 1...*n* directly coding for the factor unless the levels are to be labelled (see !L),
 - !A [*n*] is required if the data field is alphanumeric,
 - !I [*n*] is required if the data is numeric but not 1...*n*; !I must be followed by *n* if more than 1000 codes are present,
 - !L *s* is used when the data field is numeric with values 1...*n* and labels are to be assigned to the *n* levels, for example
 Sex !L Male Female
 If there are many labels, they may be written over several lines by using a trailing comma to indicate continuation of the list.
 - !P indicates the special case of a pedigree factor; ASReml will determine the levels from the pedigree file, see Section 9.3,

A warning is printed if the nominated value for *n* does not agree with the actual number of levels found in the data and if the nominated value is too small the correct value is used.

- for a group of *m* variates
 - !G *m* is used when *m* contiguous data fields are to be treated as a factor or group of variates. For example

longhand	shorthand
<pre style="margin: 0;"> X1 X2 X3 X4 X5 y data.dat y ~ mu X1 X2 X3 X4 X5</pre>	<pre style="margin: 0;"> X !G 5 y data.dat y ~ mu X</pre>
- so that the 5 variates can be referred to in the model as **X** by using `X !G 5`

- *transformations*
 - see below

5.5 Transforming the data

Transformation is the process of modifying the data (for example, dividing all of the data values in a field by 10) or deriving new data for analysis (for example, summing the data in two fields). Sometimes it will be easier to use a spreadsheet to calculate derived variables than to modify variables using ASReml transformations.

Note that in ASReml

- alpha and integer fields are converted to numbers before transformations are applied,
- transformations may be applied to any data field since all data fields are stored as real numbers, but it may not be sensible to change factor level codes,
- all fields in a record are read before transformations are performed,
- transformations are performed in the order of appearance.

Transformation syntax

Transformations have one of six forms, namely

<i>!operator</i>	to perform an operation on the current field, for example, <i>!ABS</i> to take absolute values,
<i>!operator value</i>	to perform an operation involving an argument on the current field, for example, <i>!+3</i> to add 3 to all elements in the field,
<i>!operator V field</i>	to perform an operation on the current field using the data in another field, for example, <i>!-V2</i> to subtract field 2 from the current field
<i>!V target</i>	to change the field modified by subsequent transformations,
<i>!V target = value</i>	to change all of the data in a target field to a given value,
<i>!V target = V field</i>	to overwrite the data in a target field by the data values of another field; a special case is when <i>field</i> is 0 instructing ASReml to put the record number into the target field.

- *!* flags the presence of the transformation

- *operator* is one of the symbols outlined in Table 5.1,
- *value* is the argument required by the transformation,
- *V* is a literal character and is followed by the number (*target* or *field*) of a data field; the data field is used or modified depending on the context,
- *Vfield* may be replaced by the name of the field if the field is already named.
- in the first three forms the operation is performed on the *current* field; this will be the field associated with the label unless this field has been reset by specifying a new *target* in a preceding transformation,
- the fourth form changes the focus for subsequent transformations to the field *target*,
- in the last two forms the operation is performed on the *target* field which is modified as a consequence, for example, to create a copy of field 11, which is an existing data field, we would use !V22=V11 which copies field 11 into field 22; if field 22 does not exist in the data it is created; note that subsequent transformations on the line will also apply to the *target*,
- ASReml expects to read values for each variable from the data file unless the variable and all subsequent ones are created using !=. The number of variables read can be explicitly set using the !READ qualifier described in Table 5.5.

Table 5.1: List of transformation qualifiers and their actions with examples

qualifier	argument	action	examples
!=, !+, !-, !*, !/	v	usual arithmetic meaning; note that, $0/0$ gives 0 but $v/0$ gives a missing value where v is any number other than 0.	<code>yield !/10</code>
!^	v	raises the data (which must be positive) to the power v .	<code>yield SQRYld !=yield !^0.5</code>
!^	0	takes natural logarithms of the data (which must be positive).	<code>yield LNyield !=yield !^0</code>
!^	-1	takes reciprocal of data (data must be positive).	<code>yield INVyield !=yield !^-1</code>
!>, !<, !<>, !==, !<=, !>=	v	logical operators forming 1 if true, 0 if false.	<code>yield high !=yield !>10</code>
!ABS		takes absolute values - no argument required.	<code>yield ABSyield !=yield !ABS</code>
!ARCSIN	v	forms an ArcSin transformation using the sample size specified in the argument which may be a number or another field. In the side example, for two existing fields Germ and Total containing counts, we form the ArcSin for their ratio (ASG) by copying the Germ field and applying the ArcSin transformation using the Total field as sample size.	<code>Germ Total ASG !=Germ !ARCSIN Total</code>
!COS, !SIN	s	takes cosine and sine of the data variable with period s having default 2π ; omit s if data is in radians, set s to 360 if data is in degrees.	<code>Day CosDay !=Day !COS 365</code>
!D	v	!D v removes records with v or 'missing value' in the field; if !D is used after !A or !I, v should refer to the encoded factor level rather than the value in the data file (see also Section 4.2).	<code>yield !D-9</code>
!EXP		takes antilog base e - no argument required.	<code>Rate !EXP</code>

List of transformation qualifiers and their actions with examples

qualifier	argument	action	examples
!Jddm, !Jmmd !Jyyd		!Jddm converts a date in the form <i>ddmmccyy</i> , <i>ddmmyy</i> or <i>ddmm</i> into days. !Jmmd converts a date in the form <i>ccyyymmdd</i> , <i>yymmdd</i> or <i>mmdd</i> into days. !Jyyd converts a date in the form <i>ccyyddd</i> or <i>yyddd</i> into days. These calculate the number of days since December 31 1900 and are valid for dates from January 1 1900 to December 31 2099; note that if <i>cc</i> is omitted it is taken as 19 if <i>yy</i> > 50 and 20 if <i>yy</i> < 50, the date string must be entirely numeric: characters such as / may not be present.	
!M	<i>v</i>	!M <i>v</i> converts data values of <i>v</i> to missing; if !M is used after !A or !I, <i>v</i> should refer to the encoded factor level rather than the value in the data file (see also Section 4.2).	yield !M-9
!MAX, !MIN, !MOD	<i>v</i>	the maximum, minimum and modulus of the field values and the value <i>v</i> .	yield !MAX 9
!NA	<i>v</i>	replaces any missing values in the variate with the value <i>v</i> .	Rate !NA 0
!SET	<i>vlist</i>	for <i>vlist</i> , a list of <i>n</i> values, the data values 1 . . . <i>n</i> are replaced by the corresponding element from <i>vlist</i> ; data values that are < 1 or > <i>n</i> are replaced by zero. <i>vlist</i> may run over several lines provided each incomplete line ends with a comma, i.e., a comma is used as a continuation symbol (see Other examples below).	treat !L C A B CvR !=treat !SET 1 -1 -1 group !=treat !SET 1, 2 2 3 3 4

List of transformation qualifiers and their actions with examples

qualifier	argument	action	examples
!SUB	<i>vlist</i>	replaces data values = v_i with their index i where <i>vlist</i> is a vector of n values. Data values not found in <i>vlist</i> are set to 0. <i>vlist</i> may run over several lines if necessary provided each incomplete line ends with a comma. ASReml allows for a small rounding error when matching. It may not distinguish properly if values in <i>vlist</i> only differ in the sixth decimal place (see Other examples below).	<code>year 3 !SUB 66 67 68</code>
!SEQ		replaces the data values with a sequential number starting at 1 which increments whenever the data value changes between successive records; the current field is presumed to define a factor and the number of levels in the new factor is set to the number of levels identified in this sequential process (see Other examples below). Missing values remain missing.	<code>plot !=V3 !SEQ</code>
!Vtarget=	<i>value</i>	assigns <i>value</i> to data field <i>target</i> overwriting previous contents; subsequent transformation qualifiers will operate on data field <i>target</i> .	<code>!V3=2.5</code>
	<i>Vfield</i>	assigns the contents of data field <i>field</i> to data field <i>target</i> overwriting previous contents; subsequent transformation qualifiers will operate on data field <i>target</i> . If <i>field</i> is 0 the number of the data record is inserted.	<code>!V10=V3</code> <code>!V11=block</code> <code>!V12=V0</code>

Other rules and examples

Other rules include the following

- missing values are unaffected by arithmetic operations, that is, missing values in the current or target column remain missing after the transformation has been performed except in assignment
 - `!+3` will leave missing values (`NA`, `*` and `.`) as missing,
 - `!=3` will change missing values to 3,
- multiple arithmetic operations cannot be expressed in a complex expression but must be given as separate operations that are performed in sequence as they appear, for example, `yield !-120 !*0.0333` would calculate $0.0333 * (\text{yield} - 120)$.

ASReml code	action
<code>yield !M0</code>	changes the zero entries in <code>yield</code> to missing values
<code>yield !^0</code>	takes natural logarithms of the <code>yield</code> data
<code>score !-5</code>	subtracts 5 from all values in <code>score</code>
<code>score !SET -0.5 1.5 2.5</code>	replaces data values of 1, 2 and 3 with -0.5, 1.5 and 2.5 respectively
<code>score !SUB -0.5 1.5 2.5</code>	replaces data values of -0.5, 1.5 and 2.5 with 1, 2 and 3 respectively; a data value of 1.51 would be replaced by 0 since it is not in the list or very close to a number in the list
<code>block 8</code>	in the case where
<code>variety 20</code>	- there are multiple units per plot,
<code>yield</code>	- contiguous plots have different treatments, and
<code>plot * !=variety !SEQ</code>	- the records are sorted units within plots within blocks,
	this code generates a <code>plot</code> factor assuming a new plot whenever the code in <code>V2</code> (<code>variety</code>) changes; no field would be read unless there were later definitions which are not created by transformation,
<code>Var 3</code>	assuming <code>Var</code> is coded 1:3 and <code>Nit</code> is coded 1:4, this
<code>Nit 4</code>	syntax could be used to create a new factor <code>VxN</code> with
<code>VxN 12 !=Var !-1 !*4 !+Nit</code>	the 12 levels of the composite <code>Var</code> by <code>Nit</code> factor.

Special note on covariates

Covariates are variates that appear as independent variables in the model. It is recommended that covariates be centred and scaled to have a mean of zero and a variance of approximately one to avoid failure to detect singularities. This can be achieved either

- externally to ASReml in data file preparation,
- using `!- mean !/ scale` where *mean* and *scale* are user supplied values, for example, `age !-200 !/10`.

5.6 Data file line

The purpose of the data file line is to

- nominate the data file,
- specify qualifiers to modify
 - the reading of the data,
 - the output produced,
 - the operation of ASReml , see Table 5.2.

```
NIN Alliance Trial 1989
variety !A
:
:
row 22
column 11
alliance89.asd !skip 1
yield ~ mu variety
:
:
```

Data line syntax

The data file line appears in the ASReml command file in the form

datafile [*qualifiers*]

- *datafile* is the path name of the file that contains the variates, factors, covariates, traits (response variates) and weight variables represented as data fields, see Chapter 4; the path name may not include embedded blanks,
- the *qualifiers* tell ASReml to modify either
 - the reading of the data and/or the output produced, see Table 5.2 below for a list of data file related qualifiers,
 - the operation of ASReml, see Tables 5.3 to 5.6 for a list of job control qualifiers
- the data file related qualifiers must appear on the data file line,
- the job control qualifiers may appear on the data file line or on the following line,

- the arguments to qualifiers are represented by the following symbols

f — a filename,
n — an integer number, typically a count,
p — a vector of real numbers, typically in increasing order,
r — a real number,
s — a character string,
t — a model term label,
v — the number or label of a data variable,
vlist — a list of variable labels.

5.7 Data file qualifiers

Table 5.2 lists the qualifiers relating to data input and output. These include !SKIP, !CSV, !FILTER, !SELECT and !PRINT. These must occur on the data file name line. Use the Index to check for examples or further discussion of these qualifiers.

Table 5.2: Qualifiers relating to data input and output

<i>qualifier</i>	action
Frequently used data file qualifier	
!SKIP <i>n</i>	causes the first <i>n</i> records of the (non-binary) data file to be ignored. Typically these lines contain column headings for the data fields.
Other data file qualifiers	
!CSV	used to make consecutive commas imply a missing value; this, is automatically set if the file name ends with <code>.csv</code> or <code>.CSV</code> (see also Section 4.2)
!FILTER <i>v</i> !SELECT <i>n</i>	enables a subset of the data to be analysed; <i>v</i> is the number or name of a data field. When reading data, the value in field <i>v</i> is checked <i>after</i> any transformations are performed. If <code>!select</code> is omitted, records with zero in field <i>v</i> are omitted from the analysis. Otherwise, records with <i>n</i> in field <i>v</i> are retained and all other records are omitted. Warning If the filter column contains a missing value, the value from the previous non-missing record is assumed in that position.

5.8 Job control qualifiers

The following table provides a comprehensive list of the job control qualifiers. These change or control various aspects of the analysis. Job control qualifiers may be placed on the data file line and the following line. Use the Index to check for examples or further discussion of these qualifiers.

Important Many of these are only required in very special circumstances and new users should not attempt to understand all of them. You do need to understand that all general qualifiers are specified here. Many of these qualifiers are referenced in other chapters where their purpose will be more evident.

Table 5.3: List of frequently used job control qualifiers

<i>qualifier</i>	<i>action</i>
<code>!MAXIT <i>n</i></code>	<p>sets the maximum number of iterations; the default is 10 iterations. ASReml iterates for <i>n</i> iterations unless convergence is achieved first. Convergence is presumed when the REML log-likelihood changes less than 0.002* <i>current iteration</i> number and the individual variance parameter estimates change less than 1%.</p> <p>To abort the job at the end of the current iteration, create a file named <code>ABORTASR.NOW</code> in the directory in which the job is running. At the end of each iteration, ASReml checks for this file and if present, stops the job, writing out the normal results. On case sensitive operating systems (eg. Unix), the filename must be upper case. Note that the <code>ABORTASR.NOW</code> file is deleted so nothing of importance should be placed in it.</p> <p>If the job has not converged in <i>n</i> iterations, use the <code>!CONTINUE</code> qualifier to resume iterating from the current point.</p> <p>If you perform a system level abort (CTRL C or close the program window) output files other than the <code>.rsv</code> file will be incomplete. The <code>.rsv</code> file should still be functional for resuming iteration at the most recent parameter estimates (see <code>!CONTINUE</code>).</p> <p>Use <code>!MAXIT 1</code> where you want estimates of fixed effects and predictions of random effects for the particular set of variance parameters supplied as initial values. Otherwise the estimates and predictions will be for the updated variance parameters (see also the <code>!BLUP</code> qualifier below).</p>

Table 5.4: List of occasionally used job control qualifiers

<i>qualifier</i>	action
<code>!ASMV <i>n</i></code>	<p>indicates a multivariate analysis is required although the data is presented in a univariate form. 'Multivariate Analysis' is used in the narrow sense where an unstructured error variance matrix is fitted across traits, records are independent, and observations may be missing for particular traits, see Chapter 8 for a complete discussion.</p> <p>The data is presumed arranged in lots of <i>n</i> records where <i>n</i> is the number of <i>traits</i>. It may be necessary to expand the data file to achieve this structure, inserting a missing value <code>NA</code> on the additional records. This option is sometimes relevant for some forms of repeated measures analysis. There will need to be a factor in the data to code for trait as the intrinsic <code>Trait</code> factor is undefined when the data is presented in a univariate manner.</p>
<code>!ASUV</code>	<p>indicates that a <i>univariate</i> analysis is required although the data is presented in a multivariate form. Specifically, it allows you to have an error variance other than $I \otimes \Sigma$ where Σ is the unstructured variance structure (with the <code>US</code> variance model in <code>ASReml</code>, see Table 7.3). If there are <i>missing values</i> in the data, you must include <code>!f mv</code> on the end of the linear model. The intrinsic <code>Trait</code> factor is defined and may be used in the model, see Chapter 8 for a complete discussion.</p> <p>This option is used for repeated measures analysis when the variance structure required is not the standard multivariate unstructured matrix.</p>
<code>!COLFAC <i>v</i></code>	<p>is used with <code>!SECTION <i>v</i></code> and <code>!ROWFAC <i>v</i></code> to instruct <code>ASReml</code> to set up R structures for analysing a multi-environment trial with a separable first order autoregressive model for each site (environment). <i>v</i> is the name of a factor or variate containing column numbers (1 . . . <i>n_c</i> where <i>n_c</i> is the number of columns) on which the data is to be sorted. See <code>!SECTION</code> for more detail.</p>

List of occasionally used job control qualifiers

<i>qualifier</i>	action
<code>!CONTINUE</code>	<p>is used to restart/resume iterations from the point reached in a previous iteration. This qualifier can alternately be set from the command line using the option letters C (continue) or F (final) (see Section 12.3 on job control command line options). After each iteration, ASReml writes the current values of the variance parameters to a file with extension <code>.rsv</code> (re-start values). The <code>.rsv</code> file contains information to identify individual variance parameters.</p> <p>The <code>!CONTINUE</code> qualifier causes ASReml to scan the <code>.rsv</code> file for parameter values related to the current model replacing the values obtained from the <code>.as</code> file before iteration resumes. If the model has changed, ASReml will pick up the values it recognises as being for the same terms. Furthermore, ASReml will use estimates in the <code>.rsv</code> file for certain models to provide starting values for certain more general models, inserting reasonable defaults where necessary. The transitions recognised are listed and discussed in 7.10.</p> <p>DIAG to FA1 DIAG to CORUH (uniform heterogeneous) CORUH to FA1 FAi to FA$i+1$ FAi to CORGH (full heterogeneous) FAi to US (full heterogeneous) CORGH (heterogeneous) to US See the <code>!DOPART</code> qualifier for how this may be exploited.</p>
<code>!DISPLAY n</code>	<p>is used to select particular graphic displays. In spatial analysis, four graphic displays are possible. Coding these</p> <p>1=variogram 2=histogram 4=row and column trends 8=perspective plot of residuals</p> <p>set n to the sum of the codes for the desired graphics. The default is 9. Note that the graphics are only displayed in compiled versions of ASReml linked with Interacter (that is, SUN and PC) versions. However, line printer versions of these graphics are written to the <code>.res</code> file. See the G command line option (Section 12.3 on graphics) for how to save the graphs in a file for printing.</p> <p>Use <code>!NODISPLAY</code> to suppress graphic displays.</p>

List of occasionally used job control qualifiers

<i>qualifier</i>	action
<code>!DOPART <i>n</i></code>	<p>is provided so that several analyses can be coded and run sequentially without having to edit the .as file between runs. Sections of the .as file can be delineated by <code>!PART <i>n</i></code> lines. These lines begin with the word <code>!PART</code> followed by a number. The lines after the <code>!PART</code> line are ignored unless <i>n</i> on the <code>!DOPART</code> qualifier matches <i>n</i> on the preceding <code>!PART</code> statement or there is no <i>n</i> on the <code>!PART</code> line which means always include. By setting the <code>!DOPART</code> argument to <code>\$1</code>, the part processed can be specified at run time from the command line (see Section 12.4). See Sections 15.7 and 15.10 for examples. The default value of <i>n</i> is 1.</p> <p>One situation where this might be useful is where it is necessary to run simpler models to get reasonable starting values for more complex variance models. The more complex models are specified in later parts and the <code>!CONTINUE</code> command is used to pick up the previous estimates.</p>
<code>!MVINCLUDE</code>	<p>when <code>ASReml</code> detects missing values in the design it will report this fact and abort the job unless <code>!MVINCLUDE</code> is specified. Otherwise, use the <code>!D</code> transformation to drop the records with the missing values.</p>
<code>!MVREMOVE</code>	<p>instructs <code>ASReml</code> to discard records which have missing values in the design matrix.</p>
<code>!NODISPLAY</code>	<p>used to suppress the graphic display of the variogram and residuals which is otherwise produced for spatial analyses in the PC and SUN versions. This option is usually set on the command line using the option letter N (see Section 12.3 on graphics). The display is dependent on the program being compiled with the Interacter library. The text version of the graphics is still written to the .res file.</p>
<code>!PVAL <i>v p</i></code>	<p>provides a mechanism for nominating the particular points to be predicted for covariates including <code>fac()</code>, <code>leg()</code>, <code>spl()</code> and <code>pol()</code>. The points are specified here so that they can be included in the appropriate design matrices. <i>v</i> is the name of a data field. <i>p</i> is the list of values at which prediction is required.</p>
<code>!PVAL <i>f vlist</i></code>	<p>is used to read <code>predict_points</code> for several variables from a file. <i>vlist</i> is the names of the variables having values defined. If the file contains unwanted fields, put the pseudo variate label <code>skip</code> in the appropriate position in <i>vlist</i> to ignore them. The file should only have numeric values. <code>predict_points</code> cannot be specified for design factors.</p>

List of occasionally used job control qualifiers

<i>qualifier</i>	action
<code>!ROWFAC v</code>	is used with <code>!SECTION v</code> and <code>!COLFAC v</code> to instruct ASReml to setup the R structures for multi-environment spatial analysis. <i>n</i> is the name of a factor or variate containing <i>row</i> numbers (1 ... <i>n_r</i> where <i>n_r</i> is the number of rows) on which the data is to be sorted. See <code>!SECTION</code> for more detail.
<code>!SECTION v</code>	specifies the factor in the data that defines the data sections. This qualifier enables ASReml to check that sections have been correctly dimensioned but does not cause ASReml to sort the data unless <code>!ROWFAC</code> and <code>!COLFAC</code> are also specified. Data is assumed to be sorted by section. The following is a basic example assuming 5 sites (sections). When <code>!ROWFAC v</code> and <code>!COLFAC v</code> are both specified ASReml generates the R structures for a standard $AR \otimes AR$ spatial analysis. The R structure lines that a user would normally be required to work out and type into the <code>.as</code> file (see the example of Section 15.6) are written to the <code>.res</code> file. The user may then cut and paste them into the <code>.as</code> file for a later run if the structures need to be modified. Basic multi-environment trial analysis <pre> site 5 # sites coded 1 ... 5 column * # columns coded 1 ... row * # rows coded 1 ... variety !A # variety names yield met.dat !SECTION site !ROWFAC row !COLFAC col yield ~ site !r variety site.variety !f mv site 2 0 # variance header line # asreml inserts the 10 lines required to define # the R structure lines for the five sites here </pre>

List of occasionally used job control qualifiers

<i>qualifier</i>	action
<code>!SPLINE <i>t p</i></code>	<p>defines a spline model term with an explicit set of knot points. The basic form of the spline model term, <code>spl(<i>v</i>)</code>, is defined in Table 6.1 where <i>v</i> is the underlying variate. The basic form uses the unique data values as the knot points. The extended form is <code>spl(<i>v</i>,<i>n</i>)</code> which uses <i>n</i> knot points. Use this <code>!SPLINE</code> qualifier where <i>t</i> is <code>spl(<i>v</i>,<i>n</i>)</code> to supply an explicit set of <i>n</i> knot points (<i>p</i>) for the model term <i>t</i>. Using the extended form without using this qualifier results in <i>n</i> equally spaced knot points being used. The <code>!SPLINE</code> qualifier may only be used on a line by itself after the data file line and before the model line.</p> <p>When knot points are explicitly supplied they should be in increasing order and adequately cover the range of the data or <code>ASReml</code> will modify them before they are applied. If you choose to spread them over several lines use a comma at the end of incomplete lines so that <code>ASReml</code> will continue reading values from the next line of input. If the explicit points do not adequately cover the range, a message is printed and the values are rescaled unless <code>!NOCHECK</code> is also specified. Inadequate coverage is when the explicit range does not cover the midpoint of the actual range. See also <code>!KNOTS</code>, <code>!PVAL</code> and <code>!SCALE</code>.</p>
<code>!STEP <i>r</i></code>	<p>reduces the update step sizes of the variance parameters. The default value is the reciprocal of the square root of <code>!MAXIT</code>. It may be set between 0.01 and 1.0. The step size is increased towards 1 each iteration. Starting at 0.1, the sequence would be 0.1, 0.32, 0.56, 1. This option is useful when you do not have good starting values, especially in multivariate analyses.</p>

Table 5.5: List of rarely used job control qualifiers

<i>qualifier</i>	<i>action</i>
<code>!BLUP <i>n</i></code>	<p>restricts ASReml to performing only part of the first iteration. The estimation routine is aborted after</p> <p><i>n</i> = 1: forming the estimates of the vector of fixed and random effects,</p> <p><i>n</i> = 2: forming the estimates of the vector of fixed and random effects, REML log-likelihood and residuals (this is the default),</p> <p><i>n</i> = 3: forming the estimates of the vector of fixed and random effects, REML log-likelihood, residuals and inverse coefficient matrix.</p> <p>ASReml prints its standard reports as if it had completed the iteration normally, but since it has not completed it, some of the information printed will be incorrect. In particular, variance information on the variance parameters will always be wrong. Standard errors on the estimates will be wrong unless <i>n</i>=3. Residuals are not available if <i>n</i>=1. Use of <i>n</i>=3 or <i>n</i>=2 will halve the processing time when compared to the alternative of using <code>!MAXIT 1</code>. However, <code>!MAXIT 1</code> does result in a complete and correct output report.</p>
<code>!DATAFILE <i>f</i></code>	<p>specifies the datafile name replacing the one obtained from the datafile name line. It is required when different <code>!PARTS</code> (see <code>!DOPART</code> in Table 5.4) of a job must read different files. The <code>!SKIP</code> qualifier, if specified, will be applied when reading the file.</p>
<code>!DENSE <i>n</i></code>	<p>reduces the number of equations solved densely. By default, sparse matrix methods are applied to the random effects and any fixed effects listed after random factors. Use <code>!DENSE <i>n</i></code> to apply sparse methods to effects listed before <code>!r</code>. This qualifier will not increase the number of dense equations (maximum of 800) and individual model terms will not be split so that only part is in the dense section. <i>n</i> should be kept small (<100) for faster processing.</p>
<code>!DF <i>n</i></code>	<p>alters the error degrees of freedom from ν to $\nu - n$. This qualifier might be used when analysing pre-adjusted data to reduce the degrees of freedom (<i>n</i> negative) or when weights are used in lieu of actual data records to supply error information (<i>n</i> positive). The degrees of freedom is only used in the calculation of the residual variance in a univariate single site analysis. The option will have no effect in analyses with multiple error variances (for sites or traits) other than in the reported degrees of freedom. Use <code>!ADJUST <i>r</i></code> rather than <code>!DF <i>n</i></code> if <i>r</i> is not a whole number.</p>

List of rarely used job control qualifiers

<i>qualifier</i>	action
<code>!FORMAT s</code>	<p>supplies a FORTRAN like <code>FORMAT</code> statement for reading fixed format files. A simple example is <code>!FORMAT(3I4,5F6.2)</code> which reads 3 integer fields and 5 floating point fields from the first 42 characters of each data line. A format statement is enclosed in parentheses and may include 1 level of nested parentheses, for example, e.g. <code>!FORMAT(4x,3(I4,f8.2))</code>. Field descriptors are</p> <ul style="list-style-type: none"> • <code>rX</code> to skip r character positions, • <code>rAw</code> to define r consecutive fields of w characters width, • <code>rIw</code> to define r consecutive fields of w characters width, and • <code>rFw.d</code> to define r consecutive fields of w characters width; d indicates where to insert the decimal point if it is not explicitly present in the field, <p>where r is an optional repeat count.</p> <p>In <code>ASReml</code>, the <code>A</code> and <code>I</code> field descriptors are treated identically and simply set the field width. Whether the field is interpreted alphabetically or as a number is controlled by the <code>!A</code> qualifier.</p> <p>Other legal components of a format statement are</p> <ul style="list-style-type: none"> • the <code>,</code> character; required to separate fields - blanks are not permitted in the format. • the <code>/</code> character; indicates the next field is to be read from the next line. However a <code>/</code> on the end of a format to skip a line is not honoured. • <code>BZ</code>; the default action is to read blank fields as missing values. <code>*</code> and <code>NA</code> are also honoured as missing values. If you wish to read blank fields as zeros, include the string <code>BZ</code>. • the string <code>BM</code>; switches back to 'blank missing' mode. • the string <code>Tc</code>; moves the 'last character read' pointer to line position c so that the next field starts at position $c + 1$. For example <code>T0</code> goes back to the beginning of the line. • the string <code>D</code>; invokes debug mode. <p>A format showing these components is <code>!FORMAT(D,3I4,8X,A6,3(2x,F5.2)/4x,BZ,20I1)</code> and is suitable for reading 27 fields from 2 data records such as</p> <pre>111122223333xxxxxxxxALPHAFxx 4.12xx 5.32xx 6.32 xxxx123 567 901 345 7890</pre>
<code>!G v</code>	is used to set a grouping variable for plotting, see <code>!X</code> .

List of rarely used job control qualifiers

<i>qualifier</i>	action
!JOIN	is used to join lines in plots, see !X.
!PRINT <i>n</i>	causes ASReml to print the transformed data file to <i>base-name.asp</i> . If $n < 0$, data fields $1 \dots \text{mod}(n)$ are written to the file, $n = 0$, nothing is written, $n = 1$, all data fields are written to the file if it does not exist, $n = 2$, all data fields are written to the file overwriting any previous contents, $n > 2$, data fields $n \dots t$ are written to the file where t is the last defined column.
!READ <i>n</i>	formally instructs ASReml to read n data fields from the data file. It is needed when there are extra columns in the data file that are only required for combination into earlier fields in transformations.
!RECODE	required when reading a binary data file with pedigree identifiers that have not been recoded according to the pedigree file. It is not needed when the file was formed using the !SAVE option but will be needed if formed in some other way (see also Section 4.2).
!REPORT	forces ASReml to attempt to produce the standard output report when there is a failure of the iteration algorithm. Usually no report is produced unless the algorithm has at least produced estimates for the fixed and random effects in the model. Note that residuals are not included in the output forced by this qualifier. This option is primarily intended to help debugging a job that is not converging properly.
!RESIDUALS [2]	on the data file line or following lines instructs ASReml to write the transformed data and the residuals to a binary file. The residual is the last field. The file <i>basename.srs</i> is written in single precision unless the argument is 2 in which case <i>basename.drs</i> is written in double precision. The file will not be written from a spatial analysis (two-dimensional error) when the data records have been sorted into field order because the residuals are not in the same order that the data is stored. The residual from a spatial analysis will have the units part added to it when units is also fitted. The <i>.drs</i> file could be renamed (with extension <i>.dbl</i>) and used for input in a subsequent run.

List of rarely used job control qualifiers

<i>qualifier</i>	action
<code>!SAVE <i>n</i></code>	on the data file line or following lines instructs ASReml to write the data to a binary file. The file <code>asrdata.bin</code> is written in single precision if the argument <i>n</i> is 1 or 3; <code>asrdata.dbl</code> is written in double precision if the argument <i>n</i> is 2 or 4; the data values are written before transformation if the argument is 1 or 2 and after transformation if the argument is 3 or 4. The default is single precision after transformation (see also Section 4.2).
<code>!VCC <i>n</i></code>	specifies that <i>n</i> constraints are to be applied to the variance parameters. The constraint lines occur after the G structures are defined. The constraints are described in Section 7.9. The variance header line (Section 7.4) must be present, even if only <code>0 0 0</code> indicating there are no explicit R or G structures (see also Section 7.9).
<code>!X <i>v</i> !Y <i>v</i> !G <i>v</i> !JOIN</code>	is used to plot the (transformed) data. Use <code>!X</code> to specify the <i>x</i> variable, <code>!Y</code> to specify the <i>y</i> variable and <code>!G</code> to specify a grouping variable. <code>!JOIN</code> joins the points when the <i>x</i> value increases between consecutive records. The grouping variable may be omitted for a simple scatter plot. Omit <code>!Y <i>y</i></code> produce a histogram of the <i>x</i> variable.

For example,

```
!X age !Y height !G sex
```

Note that the graphs are only produced in the graphics versions of ASReml (Section 12.3).

For multivariate repeated measures data, ASReml can plot the response profiles if the first response is nominated with the `!Y` qualifier and the following analysis is of the multivariate data. ASReml assumes the response variables are in contiguous fields and are equally spaced. For example

```
Response profiles
Treatment !A
Y1 Y2 Y3 Y4 Y5
rm.asd !Y Y1 !G Treatment !JOIN
Y1 Y2 Y3 Y4 Y5 ~ Trait Treatment Trait.Treatment
```

Table 5.6: List of very rarely used job control qualifiers

<i>qualifier</i>	action
<code>!CINV <i>n</i></code>	prints the portion of the inverse of the coefficient matrix pertaining to the n^{th} term in the linear model. Because the model has not been defined when <code>ASReml</code> reads this line, it is up to the user to count the terms in the model to identify the portion of the inverse of the coefficient matrix to be printed. The option is ignored if the portion is not wholly in the <code>SPARSE</code> stored equations. The portion of the inverse is printed to a file with extension <code>.cii</code> . The sparse form of the matrix only is printed in the form $i\ j\ C^{ij}$, that is, elements of C^{ij} that were not needed in the estimation process are not included in the file.
<code>!EXTRA <i>n</i></code>	forces another <code>mod(<i>n</i>,10)</code> rounds of iteration after apparent convergence. The default for <i>n</i> is 1. This qualifier has lower priority than <code>!MAXIT</code> and <code>ABORTASR.NOW</code> (see <code>!MAXIT</code> for details). Convergence is judged by changes in the REML log-likelihood value and variance parameters. However, sometimes the variance parameter convergence criteria has not been satisfied.
<code>!FACPOINTS <i>n</i></code>	affects the number of distinct points recognised by the <code>fac()</code> model function (Table 6.1). The default value of <i>n</i> is 1000 so that points closer than 0.1% of the range are regarded as the same point.
<code>!KNOTS <i>n</i></code>	changes the default knot points used when fitting a spline to data with more than <i>n</i> different values of the spline variable. When there are more than <i>n</i> (default 50) points, <code>ASReml</code> will default to using <i>n</i> equally spaced knot points.
<code>!NOCHECK</code>	forces <code>ASReml</code> to use any explicitly set spline knot points (see <code>!SPLINE</code>) even if they do not appear to adequately cover the data values.
<code>!NOREORDER</code>	prevents the automatic reversal of the order of the fixed terms (in the dense equations) and possible reordering of terms in the sparse equations.
<code>!NOSCRATCH</code>	forces <code>ASReml</code> to hold the data in memory. <code>ASReml</code> will usually hold the data on a scratch file rather than in memory. In large jobs, the system area where scratch files are held may not be large enough. A Unix system may put this file in the <code>/tmp</code> directory which may not have enough space to hold it.
<code>!POLPOINTS <i>n</i></code>	affects the number of distinct points recognised by the <code>pol()</code> model function (Table 6.1). The default value of <i>n</i> is 1000 so that points closer than 0.1% of the range are regarded as the same point.

List of very rarely used job control qualifiers

<i>qualifier</i>	action
<code>!PPOINTS <i>n</i></code>	influences the number of points used when predicting splines and polynomials. The design matrix generated by the <code>leg()</code> , <code>pol()</code> and <code>sp1()</code> functions are modified to include extra rows that are accessed by the <code>PREDICT</code> directive. The default value of <i>n</i> is 21 if there is no <code>!PPOINTS</code> qualifier. The range of the data is divided by <i>n-1</i> to give a step size <i>i</i> . For each point <i>p</i> in the list, a predict point is inserted at $p + i$ if there is no data value in the interval $[p, p+1.1 \times i]$. <code>!PPOINTS</code> is ignored if <code>!PVAL</code> is specified for the variable. This process also effects the number of levels identified by the <code>fac()</code> model term.
<code>!SCALE 1</code>	When forming a design matrix for the <code>sp1()</code> model term, <code>ASReml</code> uses a standardized scale (independent of the actual scale of the variable). The qualifier <code>!SCALE 1</code> forces <code>ASReml</code> to use the scale of the variable. The default standardised scale is appropriate in most circumstances.
<code>!SLOW <i>n</i></code>	reduces the update step sizes of the variance parameters more persistently than the <code>!STEP <i>r</i></code> qualifier. If specified, <code>ASReml</code> looks at the potential size of the updates and if any are large, it reduces the size of <i>r</i> . If <i>n</i> is greater than 10 <code>ASReml</code> also modifies the Information matrix by multiplying the diagonal elements by <i>n</i> . This has the effect of further reducing the updates. This option may help when you do not have good starting values, especially in multivariate analyses.

6

Command file: Specifying the terms in the mixed model

Introduction

Specifying model formulae in ASReml

General rules

Examples

Fixed terms in the model

Primary fixed terms

Sparse fixed terms

Random terms in the model

Interactions and conditional factors

Interactions

Conditional factors

Alphabetic list of model functions

Weights

Missing values

Missing values in the response

Missing values in the explanatory variables

Some technical details about model fitting in ASReml

Sparse *versus* dense

Ordering of terms in ASReml

Aliasing and singularities

Examples of aliasing

Testing of terms

6.1 Introduction

The linear mixed model is specified in ASReml as a series of model terms and qualifiers. In this chapter the model formula syntax is described.

6.2 Specifying model formulae in ASReml

The linear mixed model is specified in ASReml as a series of model terms and qualifiers. Model terms include factor and variate labels (Section 5.4), functions of labels, special terms and interactions of these. The model is specified immediately after the data file and any job control qualifier lines. The syntax for specifying the model is

```
NIN Alliance Trial 1989
variety
:
column 11
alliance89.asd !skip 1
yield ~ mu variety !r repl,
!f mv
1 2
11 column AR1 .3
22 row AR1 .3
```

response [**!wt** *weight*] ~ *fixed* [**!r** *random*] [**!f** *sparse_fixed*]

- *response* is the label for the response variable(s) to be analysed; multivariate analysis is discussed in Chapter 8,
- *weight* is a label of a variable containing weights; weighted analysis is discussed in Section 6.7,
- ~ separates *response* from the list of fixed and random terms,
- *fixed* represents the list of primary fixed explanatory terms, that is, variates, factors, interactions and special terms for which analysis of variance (ANOVA) type tests are required. See Table 6.1 for a brief definition of model terms, allowable operators and commonly used functions. The full definition is in Section 6.6,
- *random* represents the list of explanatory terms to be fitted as random effects, see Table 6.1,
- *sparse_fixed* are additional fixed terms not included in the ANOVA table.

General rules

The following general rules apply in specifying the linear mixed model

- all elements in the model must be space separated,
- the character ~ separates the response variables(s) from the explanatory variables in the model,

Choose labels that will avoid confusion

- data fields are identified in the model by their labels
 - labels are case sensitive,
 - labels may be abbreviated (truncated) when used in the model line but care must be taken that the truncated form is not ambiguous. If the truncated form matches more than one label, the term associated with the first match is assumed,
 - model terms may only appear once in the model line; repeated occurrences are ignored,
 - model terms other than the original data fields are defined the first time they appear on the model line. They may be abbreviated (truncated) if they are referred to again provided no ambiguity is introduced.
Important It is often clearer if labels are not abbreviated. If abbreviations are used then they need to be chosen to avoid confusion.
- if the model is written over several lines, all but the final line must end with a comma to indicate that the list is continued.

In Tables 6.1 and 6.2, the arguments in model term functions are represented by the following symbols

- f — the label of a data variable defined as a model factor,
- k, n — an integer number,
- r — a real number,
- t — a model term label (includes data variables),
- v, y — the label of a data variable,

Parsing of model terms in ASReml is not very sophisticated. Where a model term takes another model term as an argument, the argument must be predefined. If necessary, include the argument in the model line with a leading ‘-’ which will cause the term to be defined but not fitted.

Table 6.1: Summary of reserved words, allowable operators and functions

	model term	brief description	common usage	
			fixed	random
reserved words	<code>mu</code>	the constant term or intercept	✓	
	<code>mv</code>	a term to estimate missing values	✓	
	<code>Trait</code>	multivariate counterpart to <code>mu</code>	✓	
	<code>units</code>	forms a factor with a level for each experimental unit		✓
operators	<code>.</code> or <code>:</code>	placed between labels to specify an interaction	✓	✓
	<code>/</code>	placed between labels to create conditional terms (Section 6.5)	✓	✓
	<code>-</code>	placed before model terms to exclude them from the model	✓	✓
	<code>,</code>	placed at the end of a line to indicate that the model specification continues on the next line		
	<code>+</code>	treated as a space	✓	✓
	<code>!{ ... !}</code>	placed around some model terms when it is important the terms not be reordered (Section 6.4)		✓
commonly used functions	<code>at(<i>f</i>, <i>n</i>)</code>	condition on level <i>n</i> of factor <i>f</i>	✓	✓
	<code>fac(<i>v</i>)</code>	forms a factor from <i>v</i> with a level for each unique value in <i>v</i>		✓
	<code>fac(<i>v</i>, <i>y</i>)</code>	forms a factor with a level for each combination of values in <i>v</i> and <i>y</i>		
	<code>lin(<i>f</i>)</code>	forms a variable from the factor <i>f</i> with values equal to 1... <i>n</i> corresponding to level(1)...level(<i>n</i>) of the factor	✓	
	<code>spl(<i>v</i> [, <i>k</i>])</code>	forms the design matrix for the random component of a cubic spline for variable <i>v</i>		✓
other functions	<code>and(<i>t</i> [, <i>r</i>])</code>	adds <i>r</i> times the design matrix for model term <i>t</i> to the previous design matrix; <i>r</i> has a default value of 1. If <i>t</i> is complex it may be necessary to predefine it by saying <code>-t and(t, r)</code>		

Summary of reserved words, allowable operators and functions

model term	brief description	common usage	
		fixed	random
<code>cos(v, r)</code>	forms cosine from v with period r	✓	
<code>giv(f, n)</code>	associates the n th <code>.giv</code> G-inverse with the factor f		✓
<code>ide(f)</code>	fits pedigree factor f without relationship matrix		✓
<code>inv(v[, r])</code>	forms reciprocal of $v + r$	✓	
<code>leg(v, [-] n)</code>	forms $n+1$ legendre polynomials of order 0 (intercept), 1 (linear)... n from the values in v ; the intercept polynomial is omitted if v is preceded by the negative sign.	✓	
<code>log(v[, r])</code>	forms natural logarithm of $v + r$	✓	
<code>ma1(f)</code>	constructs MA1 design matrix for factor f		✓
<code>ma1</code>	forms an MA1 design matrix from plot numbers		✓
<code>pol(v, [-] n)</code>	forms $n+1$ orthogonal polynomials of order 0 (intercept), 1 (linear)... n from the values in v ; the intercept polynomial is omitted if n is preceded by the negative sign.	✓	
<code>sin(v, r)</code>	forms sine from v with period r	✓	
<code>sqrt(v[, r])</code>	forms square root of $v + r$	✓	
<code>uni(f)</code>	forms a factor with a level for each record where factor f is non-zero		✓
<code>uni(f, n)</code>	forms a factor with a level for each record where factor f has level n		✓
<code>xfa(f, k)</code>	is formally a copy of factor f with k extra levels. This is used when fitting extended factor analytic models (XFA, Table 7.3) of order k in a sparser form.		✓

Examples

ASReml code	action
<code>yield ~ mu variety</code>	fits a model with a constant and fixed variety effects
<code>yield ~ mu variety !r block</code>	fits a model with a constant term, fixed variety effects and random block effects
<code>yield ~ mu time variety time.variety</code>	fits a saturated model with fixed time and variety main effects and time by variety interaction effects
<code>livewt ~ mu breed sex breed.sex !r sire</code>	fits a model with fixed breed, sex and breed by sex interaction effects and random sire effects

6.3 Fixed terms in the model

Primary fixed terms

The *fixed* list in the model formula

- describes the fixed covariates, factors and interactions including special functions to be included in the ANOVA table,
- generally begins with the reserved word `mu` which fits a constant term, mean or intercept, see Table 6.1.

```
NIN Alliance Trial 1989
  variety
  :
  :
  row 22
  column 11
  alliance89.asd !skip 1
  !mvinclude
  yield ~ mu variety !r repl,
  !f mv
  1 2
  11 column AR1 .3
  22 row AR1 .3
```

Table 6.1 includes the reserved words and special functions used in the *fixed* list.

Sparse fixed terms

The `!f` *sparse-fixed* terms in model formula

- are the fixed covariates (for example, the fixed `lin(row)` covariate now included in the model formula), factors and interactions including special functions and reserved words (for example `mv`, see Table 6.1) for which ANOVA type tests are not required,
- include large (>100 levels) terms.

```
NIN Alliance Trial 1989
variety
:
:
row 22
column 11
alliance89.asd !skip 1
yield ~ mu variety !r repl,
!f mv lin(row)
1 2
11 column AR1 .424
22 row AR1 .904
```

6.4 Random terms in the model

The `!r` *random* terms in the model formula

- comprise random covariates, factors and interactions including special functions and reserved words, see Table 6.1,
- involve an initial non-zero variance component or ratio (relative to the residual variance) default 0.1; the initial value can be specified after the model term or if the variance structure is not scaled identity, by syntax described in detail in Chapter 7,
- an initial value of its variance (ratio) may be followed by a `!GP` (restricted to be positive, the default), `!GU` (unrestricted) or `!GF` (fixed) qualifier, see Table 7.4,
- use `!{` and `!}` to group model terms that may not be reordered. Normally ASReml will reorder the model terms in the sparse equations - putting smaller terms first to speed up calculations. However, the order must be preserved if the user defines a structure for a term which also covers the following term(s) (a way of defining a covariance structure across model terms). Grouping is specifically required if the model terms are of differing sizes (number of effects). For example, for traits `weaning_weight` and `yearling_weight`, an animal model with maternal weaning weight should specify model terms `!{ Trait.animal at(Trait,1).dam !}` when fitting a genetic covariance between the direct and maternal effects.

```
NIN Alliance Trial 1989
variety
:
:
row 22
column 11
alliance89.asd !skip 1
yield ~ mu variety !r repl,
!f mv 1 2
11 column AR1 .424
22 row AR1 .904
```

6.5 Interactions and conditional factors

Interactions

- interactions are formed by joining two or more terms with a '.' or a ':', for example, `a.b` is the interaction of factors `a` and `b`,
- interaction levels are arranged with the levels of the second factor nested within the levels of the first,
- labels of factors including interactions are restricted to 31 characters of which only the first 20 are ever displayed. Thus for interaction terms it is often necessary to shorten the names of the component factors in a systematic way, for example, if `Time` and `Treatment` are defined in this order, the interaction between `Time` and `Treatment` could be specified in the model as `Time.Treat`; remember that the first match is taken so that if the label of each field begins with a different letter, the first letter is sufficient to identify the term,
- interactions can involve model functions,
- `a*b` is expanded to `a b a.b`
- `a*b*c*d` is expanded to `a b c d a.b a.c a.d b.c b.d c.d a.b.c a.b.d a.c.d b.c.d a.b.c.d`
- `a.(b c d) e` is expanded to `a.b a.c a.d e`. This syntax is detected by the string `.(` and the closing parenthesis must occur on the same line and before any comma indicating continuation. Any number of terms may be enclosed. Each may have `-` prepended to suppress it from the model but insert an extra space before the first term if this facility is required. Each enclosed term may have initial values and qualifiers following. For example,

```
yield~site site.(lin(row) !r variety) at(site,1).(row .3 col .2)
```

expands to

```
yield~site site.lin(row) !r site.variety at(site,1).row .3,
at(site,1).col .2
```

Conditional factors

A conditional factor is a factor that is present only when another factor has a particular level.

- individual components can be specified using the `at(f, n)` function (see Table 6.2), for example, `at(site,1).row` will fit `row` as a factor only for site 1,
- a complete set of conditional terms are specified by joining two terms with a /

- `a/b` creates a series of factors representing `b` nested within `a`. `b` may be any previously defined model term. A factor is created for each level of `a`; each has the size of `b`. For example, if `site` and `geno` are factors with 3 and 10 levels respectively, then for `site/geno` ASReml constructs 3 new factors `site_1.geno`, `site_2.geno` and `site_3.geno`, each with 10 levels,
- the first term must be a factor,
- this is similar to forming an interaction except that a separate model term is created for each level of the first factor; this is useful for random terms when each component can have a different variance. The same effect is achieved by using an interaction and associating a `DIAG` variance structure with the first component (see Section 7.5).

6.6 Alphabetic list of model functions

Table 6.2 presents detailed descriptions of the model functions discussed above. Note that some three letter function names may be abbreviated to the first letter.

Table 6.2: Alphabetic list of model functions and descriptions

model function	action
<code>and(t, r)</code> <code>a(t, r)</code>	overlays (adds) r times the design matrix for model term t to the existing design matrix. Specifically, if the model up to this point has p effects and t has a effects, the a columns of the design matrix for t are multiplied by r and added to the last a of the p columns already defined. This can be used to force a correlation of 1 between two terms with different variances as in a diallel analysis <code>male and(female)</code> Note that if the overlaid term is complex, it must be predefined.
<code>at(f, n)</code>	defines a binary variable which is 1 if the factor f has level n for the record. For example, to fit a row factor only for site 3, use the expression <code>at(site,3).row</code> .
<code>cos(v, r)</code>	forms cosine from v with period r
<code>fac(v)</code> <code>fac(v, y)</code>	<code>fac(v)</code> forms a factor with a level for each value of x and any additional points inserted as discussed with the qualifiers <code>!PPOINTS</code> and <code>!PVAL</code> . <code>fac(v, y)</code> forms a factor with a level for each combination of values from v and y . The values are reported in the <code>.res</code> file.
<code>giv(f, n)</code> <code>g(f, n)</code>	associates the n th <code>.giv</code> G-inverse with the factor. This is used when there is a known (except for scale) G-structure other than the additive inverse genetic relationship matrix. The G-inverse is supplied in a file whose name has the file extension <code>.giv</code> described in Section 9.6

Alphabetic list of model functions and descriptions

model function	action
<code>ide(f)</code> <code>i(f)</code>	is used to take a copy of a pedigree factor f and fit it without the genetic relationship covariance. This facilitates fitting a <i>second animal effect</i> . Thus, to form a direct, maternal genetic and maternal environment model, the maternal environment is defined as a second animal effect coded the same as dams.
<code>inv(v[,r])</code>	forms the reciprocal of $v + r$. This may also be used to transform the response variable.
<code>leg(v, [-]n)</code>	forms $n+1$ legendre polynomials of order 0 (intercept), 1 (linear) . . . n from the values in v ; the intercept polynomial is omitted if n is preceded by the negative sign.
<code>lin(f)</code> <code>l(f)</code>	takes the coding of factor f as a covariate. The function is defined for f being a simple factor, Trait and units . The <code>lin(f)</code> function does not centre or scale the variable. Motivation: Sometimes you may wish to fit a covariate as a random factor as well. If the coding is say $1 . . n$, then you should define the field as a factor in the field definition and use the <code>lin()</code> function to include it as a covariate in the model. Do not centre the field in this case. If the covariate values are irregular, you would leave the field as a covariate and use the <code>fac()</code> function to derive a factor version.
<code>log(v[,r])</code>	forms the natural log of $v + r$. This may also be used to transform the response variable.
<code>ma1</code> <code>ma1(f)</code>	creates a first-differenced (by rows) design matrix which, when defining a random effect, is equivalent to fitting a moving average variance structure in one dimension. In the <code>ma1</code> form, the first-difference operator is coded across all data points (assuming they are in time/space order). Otherwise the coding is based on the codes in the field indicated.
<code>mu</code>	is used to fit the intercept/constant term. It is normally present and listed first in the model. It should be present in the model if there are no other fixed factors or if all fixed terms are covariates or contrasts except in the special case of regression through the origin.

Alphabetic list of model functions and descriptions

model function	action
<code>mv</code>	<p>is used to estimate missing values in the response variable. Formally this creates a model term with a column for each missing value. Each column contains zeros except for a solitary -1 in the record containing the corresponding missing value. This is used in spatial analyses so that computing advantages arising from a balanced spatial layout can be exploited. The equations for <code>mv</code> and any terms that follow are always included in the sparse set of equations.</p> <p>Missing values are handled in three possible ways during analysis. In the simplest case, records containing missing values in the response variable are deleted. For multivariate (including some repeated measures) analysis, records with missing values are not deleted but <code>ASReml</code> drops the missing observation and uses the appropriate unstructured R-inverse matrix. For regular spatial analysis, we prefer to retain separability and therefore estimate the missing value(s) by including the special term <code>mv</code> in the model.</p>
<code>pol(v,n)</code> <code>p(v,n)</code>	<p>forms a set of orthogonal polynomials of order n based on the unique values in variate (or factor) v and any additional interpolated points, see <code>!PPOINTS</code> and <code>!PVAL</code> in Table 5.4. It includes the intercept if n is positive, omits it if n is negative. For example, <code>pol(time,2)</code> forms a design matrix with three columns of the orthogonal polynomial of degree 2 from the variable <code>time</code>. Alternatively, <code>pol(time,-2)</code> is a term with two columns having centred and scaled linear coefficients in the first column and centred and scaled quadratic coefficients in the second column. The actual values of the coefficients are written to the <code>.res</code> file. This factor could be interacted with a design factor to fit random regression models.</p>
<code>sin(v,r)</code>	<p>forms sine from v with period r</p>
<code>spl(v[,k])</code> <code>s(v[,k])</code>	<p>In order to fit spline models associated with a variate v and k knot points in <code>ASReml</code>, v is included as a covariate in the model and <code>spl(v,k)</code> as a random term. The knot points can be explicitly specified using the <code>!SPLINE</code> qualifier (Table 5.4). If k is specified but <code>!SPLINE</code> is not specified, equally spaced points are used. If k is not specified and there are less than 50 unique data values, they are used as knot points. If there are more than 50 unique points then 50 equally spaced points will be used. The spline design matrix formed is written to the <code>.res</code> file. An example of the use of <code>spl()</code> is</p> <pre>price ~ mu week !r spl(week)</pre>
<code>sqrt(v[,r])</code>	<p>forms the square root of $v + r$. This may also be used to transform the response variable.</p>

Alphabetic list of model functions and descriptions

model function	action
<code>Trait</code>	is used with multivariate data to fit the individual trait means. It is formally equivalent to <code>mu</code> but <code>Trait</code> is a more natural label for use with multivariate data. It is interacted with other factors to estimate their effects for all traits.
<code>units</code>	creates a factor with a level for every record in the data file. This is used to fit the 'nugget' variance when a correlation structure is applied to the residual.
<code>uni(f[,0[,n]])</code>	creates a factor with a new level whenever there is a level present for the factor <i>f</i> . Levels (effects) are not created if the level of factor <i>f</i> is 0, missing or negative. The size may be set in the third argument by setting the second argument to zero.
<code>uni(f,k[,n])</code>	creates a factor with a level for every record subject to the factor level of <i>f</i> equalling <i>k</i> , i.e. a new level is created for the factor whenever a new record is encountered whose integer truncated data value from data field <i>f</i> is <i>k</i> . Thus <code>uni(site,2)</code> would be used to create an independent error term for site 2 in a multi-environment trial and is equivalent to <code>at(site,2).units</code> . The default size of this model term is the number of data records. The user may specify a lower number as the third argument. There is little computational penalty from the default but the <code>.sln</code> file may be substantially larger than needed.
<code>xfa(f,k)</code>	Factor analytic models are discussed in Chapter 7. There are three forms, <code>FAk</code> , <code>FACVk</code> and <code>XFAk</code> where <i>k</i> is the number of factors. The <code>XFAk</code> form is a sparse formulation that requires an extra <i>k</i> levels to be inserted into the mixed model equations for the <i>k</i> factors. This is achieved by the <code>xfa(f,k)</code> model function which defines a design matrix based on the design matrix for <i>f</i> augmented with <i>k</i> columns of zeros for the <i>k</i> factors.

6.7 Weights

Weighted analyses are achieved by using `!WT weight` as a qualifier to the response variable. An example of this is `y !WT wt ~ mu A X` where *y* is the name of the response variable and *wt* is the name of a variate in the data containing weights. If these are relative weights (to be scaled by the `units` variance) then this is all that is required. If they are absolute weights, that is, the reciprocal of known variances, use the `!S2==1` qualifier described in Table 7.4 to fix the unit variance. When a structure is present in the residuals the weights are applied as a matrix

product. If Σ is the structure and \mathbf{W} is the diagonal matrix constructed from the square root of the values of the variate weight, then $\mathbf{R}^{-1} = \mathbf{W}\Sigma^{-1}\mathbf{W}$. Negative weights are treated as zeros.

6.8 Missing values

Missing values in the response

`mv` controls the estimation of missing values. It is sometimes computationally convenient to estimate missing values, for example, in spatial analysis of regular arrays, see example **3a** in Section 7.3. The default action is to exclude units with missing values, and in multivariate analyses, the \mathbf{R} matrix is modified to reflect the missing value pattern.

Formally, `mv` creates a factor with a covariate for each missing value. The covariates are coded 0 except in the record where the particular missing value occurs, where it is coded -1.

```
NIN Alliance Trial 1989
variety
:
row 22
column 11
alliance89.asd !skip 1
yield ~ mu variety !r repl,
!f mv
1 2
11 column AR1 .424
22 row AR1 .904
```

Missing values in the explanatory variables

ASReml will abort the analysis if it finds missing values in the design matrix unless `!MVINCLUDE` or `!MVREMOVE` is specified, see Section 5.8. `!MVINCLUDE` causes the missing value to be treated as a zero. `!MVREMOVE` causes ASReml to discard the whole record. Records with missing values in particular fields can be explicitly dropped using the `!D` transformation, Table 5.1.

Covariates: Treating missing values as zero in covariates is usually only sensible if the covariate is centred (has mean of zero).

Design factors: Where the factor level is zero (or missing and the `!MVINCLUDE` qualifier is specified), no level is assigned to the factor for that record.

6.9 Some technical details about model fitting in ASReml

Sparse versus dense

ASReml partitions the terms in the linear model into two parts: a *dense* set and a *sparse* set. The partition is at the `!r` point unless explicitly set with the `!DENSE` data line qualifier or `mv` is included before `!r`, see Table 5.5. The special term `mv` is always included in sparse. Thus *random* and *sparse* terms are estimated using sparse matrix methods which result in faster processing. The inverse coefficient matrix is fully formed for the terms in the dense set. The inverse coefficient matrix is only partially formed for terms in the sparse set. Typically, the sparse set is large resulting in savings in memory and computing. A consequence is that the variance matrix for estimates is only available for equations in the dense portion.

Ordering of terms in ASReml

The order in which estimates for the fixed and random effects in linear mixed model are reported will usually differ from the order the model terms are specified. Estimates are obtained by solving the corresponding mixed model equations (Gilmour *et al.*, 1995). ASReml partitions the equations into a *dense* set which typically includes the *primary fixed* factors, and a *sparse* set which includes the rest including any missing value (`mv`) covariates. ASReml first orders the equations in the sparse part to maintain as much sparsity as it can during the estimation. After absorbing them, it absorbs the model terms associated with the dense equations in the order specified. Within each term it absorbs the terms starting with the highest level. In the dense equations terms should be specified in the ‘natural’ order of main effects before interactions so that the ANOVA table has correct marginalization.

Aliasing and singularities

A singularity is reported in ASReml when there is either

- a linear dependence in the design matrix and therefore no information left to estimate the corresponding effect, or
- no data for that fixed effect.

ASReml handles singularities by setting to zero and ignoring equations detected as aliases. Since ASReml solves from the bottom up, the first level (and hence the last level processed) of a factor is the one that will be declared singular and

dropped from the model. The number of singularities is reported in the `.asr` file immediately prior to the REML log-likelihood (LogL) line for that iteration (see Section 13.3). The singular effects can be identified in the `.sln` file (see also Section 13.3). Estimates that are zero and have zero for their standard error are the effects reported as singular.

Warning

Singularities in the *sparse-fixed* terms of the model may change with changes in the random terms included in the model. If this happens it will mean that changes in the REML log-likelihood are not valid for testing the changes made to the random model. This situation is not easily detected as the only evidence will be in the `.sln` file where different fixed effects are singular. A likelihood ratio test is not valid if the fixed model has changed.

Examples of aliasing

The sequence of models in Table 6.3 are presented to facilitate an understanding of over-parameterised models. It is assumed that `var` is a factor with 4 levels, `trt` with 3 levels and `rep` with 3 levels and that all `var.trt` combinations are present in the data.

Table 6.3: Examples of aliasing in ASReml

model	number of singularities	order of fitting
<code>yield ~ var !r rep</code>	0	<code>rep var</code>
<code>yield ~ mu var !r rep</code>	1	<code>rep mu var</code> first level of <code>var</code> is aliased and set to zero
<code>yield ~ var trt !r rep</code>	1	<code>rep var trt</code> <code>var</code> fully fitted, first level of <code>trt</code> is aliased and set to zero
<code>yield ~ mu var trt, var.trt !r rep</code>	8	<code>rep mu var trt var.trt</code> first levels of both <code>var</code> and <code>trt</code> are aliased and set to zero, together with subsequent interactions
<code>yield ~ mu var trt !r rep, !f var.trt</code>	8	<code>[var.trt rep] mu var trt</code> <code>var.trt</code> fitted before <code>mu</code> , <code>var</code> and <code>trt</code> , <code>var.trt</code> fully fitted; <code>mu</code> , <code>var</code> and <code>trt</code> are completely singular and set to zero. The order within <code>[var.trt rep]</code> is determined internally.

6.10 Testing of terms

An analysis of variance table of F ratio statistics labelled `F-incr` is displayed. It allows approximate testing of the term adjusted for terms not in the table or higher in the table as portrayed in Table 6.4.

No denominator degrees of freedom is supplied since the denominator degrees of freedom is not easy to deduce in a mixed model. The *Error degrees of freedom* is a maximum value for the denominator but sometimes a much smaller value is appropriate (Kenward and Roger, 1997).

Table 6.4: Schematic analysis of variance table from fitting the model
`yield ~ mu A B X A.B A.X B.X A.B.X !r rep`

source	F-incremental
mu	mu rep
A	A mu rep
B	B A mu rep
X	X B A mu rep
A.B	A.B X B A mu rep
A.X	A.X A.B X B A mu rep
B.X	B.X A.X A.B X B A mu rep
A.B.X	A.B.X B.X A.X A.B X B A mu rep

7

Command file: Specifying the variance structures

Introduction

Non-singular variance matrices

Variance model specification in ASReml

A sequence of structures for the NIN data

Variance structures

General syntax

Variance header line

R structure definition

G structure header and definition lines

Variance model description

Forming the variance models from correlation models

Additional notes of variance models

Variance structure qualifiers

Rules for combining variance models

G structures involving more than one random term

Constraining variance parameters

Parameter constraint within a variance model

Constraints between and within variance models

Model building using the !CONTINUE qualifier

7.1 Introduction

The subject of this chapter is variance model specification in ASReml. ASReml allows a wide range of models to be fitted. The key concepts you need to understand are

- the mixed linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\tau} + \mathbf{Z}\mathbf{u} + \mathbf{e}$ has a residual term $\mathbf{e} \sim N(\mathbf{0}, \mathbf{R})$ and random effects $\mathbf{u} \sim N(\mathbf{0}, \mathbf{G})$,
- we use the terms R structure and G structure to refer to the independent blocks of R and G respectively,
- R and G structures are typically formed as a direct product of particular variance models,
- variance models may be correlation matrices or variance matrices with equal or unequal variances on the diagonal. A model for a correlation matrix (eg. AR1) can be converted to an equal variance form (eg. AR1V) and to a heterogeneous variance form (eg. AR1H),
- variances are sometimes estimated as variance ratios (relative to the residual variance).

These issues are fully discussed in Chapter 2. In this chapter we begin by considering an ordered sequence of variance structures for the NIN variety trial (see Section 7.3). This is to introduce variance modelling in practice. We then present the topics in detail.

Non singular variance matrices

When undertaking the REML estimation, ASReml needs to invert each variance matrix. For this it requires that the matrices be negative definite or positive definite. They must not be singular. Negative definite matrices will have negative diagonal elements on the diagonal of the matrix and/or its inverse. The exception is the XFA model which has been specifically designed to fit singular matrices (Thompson *et al.* 2003).

Let $\mathbf{x}'\mathbf{A}\mathbf{x}$ represent an arbitrary quadratic form for $\mathbf{x} = (x_1, \dots, x_n)'$. The quadratic form is said to be nonnegative definite if $\mathbf{x}'\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbf{R}^n$. If $\mathbf{x}'\mathbf{A}\mathbf{x}$ is nonnegative definite and in addition the null vector $\mathbf{0}$ is the only value of \mathbf{x} for which $\mathbf{x}'\mathbf{A}\mathbf{x} = 0$, then the quadratic form is said to be positive definite. Hence the matrix \mathbf{A} is said to be positive definite if $\mathbf{x}'\mathbf{A}\mathbf{x}$ is positive definite, see Harville (1997), pp 211.

7.2 Variance model specification in ASReml

The variance models are specified in the ASReml command file after the model line, as shown in the code box. In this case just one variance model is specified (for replicates, see model **2b** below for details). `predict` and `tabulate` lines may appear after the model line and before the first variance structure line. These are described in Chapter 10.

```
NIN Alliance Trial 1989
variety !A
:
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
0 0 1
repl 1
repl 0 IDV 0.1
```

Table 7.3 presents the full range of variance models available in ASReml. The identifiers for specifying the individual variance models in the command file are described in Section 7.5 under Specifying the variance models in ASReml. Many of the models are correlation models. However, these are generalised to homogeneous variance models by appending V to the base identifier. They are generalised to heterogeneous variance models by appending H to the base identifier.

7.3 A sequence of structures for the NIN data

Eight variance structures of increasing complexity are now considered for the NIN field trial data (see Chapter 3 for an introduction to these data). This is to give you a feel for variance modelling in ASReml and some of the models that are possible. Table 7.3 lists the variance models that are available in ASReml.

[See Section 2.1](#) Before proceeding, it is useful to link this section to the algebra of Chapter 2. In this case the mixed linear model is

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

where \mathbf{y} is the vector of yield data, $\boldsymbol{\tau}$ is a vector of fixed variety effects but would also include fixed replicate effects in a simple RCB analysis and might also include fixed missing value effects when spatial models are considered, $\mathbf{u} \sim N(\mathbf{0}, \mathbf{G})$ is a vector of random effects (for example, random replicate effects) and the errors are in $\mathbf{e} \sim N(\mathbf{0}, \mathbf{R})$. The focus of this discussion is on

- changes to \mathbf{u} and \mathbf{e} and the assumptions about these terms,
- the impact this has on the specification of the G structures for \mathbf{u} and the R structures for \mathbf{e} .

1 Traditional randomised complete block (RCB) analysis

The only random term in a traditional RCB analysis of these data is the (residual) error term $\mathbf{e} \sim N(\mathbf{0}, \sigma_e^2 \mathbf{I}_{224})$. The model therefore involves just one R structure and no G structures ($\mathbf{u} = \mathbf{0}$). In ASReml

- the error term is implicit in the model and is not formally specified on the model line,
- the IID variance structure ($\mathbf{R} = \sigma_e^2 \mathbf{I}_{224}$) is the default for error.

```
NIN Alliance Trial 1989
variety !A
id
pid
raw
repl 4
:
:
row 22
column 11
nin89.asd !skip 1
yield ~ mu variety repl
```

Important The error term is *always* present in the model but does not need to be formally declared when it has the default IID structure.

2a Random effects RCB analysis

The random effects RCB model has 2 random terms to indicate that the total variation in the data is comprised of 2 components, a random replicate effect $\mathbf{u}_r \sim N(\mathbf{0}, \gamma_r \sigma_e^2 \mathbf{I}_4)$ where $\gamma_r = \sigma_r^2 / \sigma_e^2$, and error as in **1**. This model involves both the original implicit IID R structure and an implicit IID G structure for the random replicates. In ASReml

- IID variance structure is the default for the extra random terms in the model.

```
NIN Alliance Trial 1989
variety !A
id
pid
raw
repl 4
:
:
row 22
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
```

For this reason the only change to the original command file is the **!r** before repl. **Important** All random terms (other than error which is implicit) must be written after **!r** in the model specification line(s). [See Section 6.4](#)

2b Random effects RCB analysis with a G structure specified

This model is equivalent to **2a** but we explicitly specify the G structure for `repl`, that is, $\mathbf{u}_r \sim N(\mathbf{0}, \gamma_r \sigma_e^2 \mathbf{I}_4)$, to introduce the syntax.

See Section 7.4

The `0 0 1` line is called the *variance header* line. In general, the first two elements of this line refer to the R structures and the third element is the number of G structures. In this case `0 0` tells ASReml that there are no explicit R structures but there is one G structure (1). The next two lines define the G structure. The first line, a *G structure header line*, links the structure that follows to a term in the linear model (`repl`) and indicates that it involves

See page 92

one variance model (1) (a 2 would mean that the structure was the direct product of two variance models). The second line tells ASReml that the variance model for replicates is IDV of order 4 ($\sigma_r^2 \mathbf{I}_4$). The `0.1` is a starting value for $\gamma_r = \sigma_r^2 / \sigma_e^2$; a starting value must be specified. Finally, the second element (0) on the last line of the file indicates that the effects are in standard order. There is almost always a 0 (no sorting) in this position for G structures. The following points should be noted:

```
NIN Alliance Trial 1989
variety !A
id
pid
raw
repl 4
:
:
row 22
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
0 0 1
repl 1
4 0 IDV 0.1
```

- the 4 on the final line could have been written as `repl` to give

```
repl 0 IDV 0.1
```

This would tell ASReml that the order or dimension of the IDV variance model is equal to the number of levels in `repl` (4 in this case),

- when specifying G structures, the user should ensure that one scale parameter is present. ASReml does not automatically include and estimate a scale parameter for a G structure when the explicit G structure does not include one. For this reason
 - the model supplied when the G structure involves just one variance model must *not* be a correlation model (all diagonal elements equal 1),
 - all but one* of the models supplied when the G structure involves more than one variance model *must* be correlation models; the other must be either an homogeneous or a heterogeneous variance model (see Section 7.5 for the distinction between these models; see also **5** for an example),
- an initial value must be supplied for all parameters in G structure definitions. ASReml expects initial values immediately after the variance model identifier

See Sections 2.1 and 7.5

or on the next line (0.1 directly after IDV in this case),

- 0 is ignored as an initial value on the model line,
- if ASReml does not find an initial value after the identifier, it will look on the next line,
- if ASReml does not find an initial value it will stop and give an error message in the .asr file,

See Chapter 14

- in this case $\mathbf{V} = \sigma_r^2 \mathbf{Z}_r \mathbf{Z}_r' + \sigma_e^2 \mathbf{I}_{224}$ which is fitted as $\sigma_e^2 (\gamma_r \mathbf{Z}_r \mathbf{Z}_r' + \mathbf{I}_{224})$ where γ_r is a variance ratio ($\gamma_r = \sigma_r^2 / \sigma_e^2$) and σ_e^2 is the scale parameter. Thus 0.1 is a reasonable initial value for γ_r regardless of the scale of the data.

3a Two-dimensional spatial model with spatial correlation in one direction

This syntax specifies a two-dimensional spatial structure for error but with spatial correlation in the row direction only, that is, $e \sim N(\mathbf{0}, \sigma_e^2 \mathbf{I}_{11} \otimes \Sigma_r(\rho_r))$. The variance header line tells ASReml that there is one R structure (1) which is a direct product of two variance models (2); there are no G structures (0). The next two lines define the two models that form the R structure. A structure definition line must be specified for *each* model. For $\mathbf{V} = \sigma_e^2 \mathbf{I}_{11} \otimes \Sigma(\rho_r)$, the first matrix is an identity matrix of order 11 for columns (ID), the second matrix is a first order autoregressive correlation matrix of order 22 for rows (AR1) and the variance scale parameter σ_e^2 is implicit. Note the following:

See page 90

```
NIN Alliance Trial 1989
variety !A
id
pid
row
repl 4
:
row 22
column 11
nin89aug.asd !skip 1
yield ~ mu variety !f mv
1 2 0
11 column ID
22 row AR1 0.3
```

- placing `column` and `row` in the second position on lines 1 and 2 respectively tells ASReml to internally sort the data *rows within columns* before processing the job. This is to ensure that the data matches the direct product structure specified. If `column` and `row` were replaced with 0 in these two lines, ASReml would assume that the data were already sorted in this order (which is not true in this case),
- the 0.3 on line 2 specifies a starting autoregressive row correlation of 0.3. Note that for spatial analysis in two dimensions using a separable model, a complete matrix or array of plots must be present. To achieve this we augmented the data with the 18 records for the missing yields as shown on page 23. In the

augmented data file the yield data for the missing plots have all been made NA (one of the missing value indicators in ASReml) and `variety` has been arbitrarily coded LANCER for all of the missing plots (any of the variety names could have been used),

- `!f mv` is now included in the model specification. This tells ASReml to estimate the missing values. The `!f` before `mv` indicates that the missing values are fixed effects in the *sparse* set of terms,

See Chapter 13
See Sections 6.3
and 6.9

- unlike the case with G structures, ASReml automatically includes and estimates a scale parameter for R structures (σ_e^2 for $\mathbf{V} = \sigma_e^2 (\mathbf{I}_{11} \otimes \boldsymbol{\Sigma}(\rho_r))$ in this case). This is why the variance models specified for `row` (AR1) and `column` (ID) are correlation models. The user could specify a non-correlation model (diagonal elements $\neq 1$) in the R structure definition, for example, ID could be replaced by IDV to represent $\mathbf{V} = \sigma_e^2 (\sigma_c^2 \mathbf{I}_{11}) \otimes \boldsymbol{\Sigma}(\rho_r)$. However, IDV would then need to be followed by `!S2==1` to fix σ_e^2 at 1 and prevent ASReml trying (unsuccessfully) to estimate two confounded parameters: the scale parameter associated with IDV and the implicit error variance parameter, see Section 2.1 under Combining variance models. Specifically, the code

See Sections 2.1
and 7.5

See Section 7.7

```
11 column IDV 48 !S2==1
```

would be required in this case, where 48 is the starting value for the variances. This complexity allows for heterogeneous error variance.

3b Two-dimensional separable autoregressive spatial model

This model extends **3a** by specifying a first order autoregressive correlation model of order 11 for columns (AR1). The R structure in this case is therefore the direct product of two autoregressive correlation matrices that is, $\mathbf{V} = \sigma_e^2 \boldsymbol{\Sigma}_c(\rho_c) \otimes \boldsymbol{\Sigma}_r(\rho_r)$, giving a two-dimensional first order separable autoregressive spatial structure for error. The starting column correlation in this case is also 0.3. Again note that σ_e^2 is implicit.

```
NIN Alliance Trial 1989
variety !A
id
:
:
row 22
column 11
nin89aug.asd !skip 1
yield ~ mu variety !f mv
1 2 0
11 column AR1 0.3
22 row AR1 0.3
```

3c Two-dimensional separable autoregressive spatial model with measurement error

This model extends **3b** by adding a random `units` term. Thus

$\mathbf{V} = \sigma_e^2 (\gamma_\eta \mathbf{I}_{242} + \boldsymbol{\Sigma}_c(\rho_c) \otimes \boldsymbol{\Sigma}_r(\rho_r))$. The reserved word `units` tells ASReml to construct an additional random term with one level for each experimental unit so that a second (independent) error term can be fitted. A `units` term is fitted in the model in cases like this, where a variance structure is applied to the errors. Because a G structure is not explicitly specified here for `units`, the default IDV structure is assumed. The `units` term is often fitted in spatial models for field trial data to allow for a nugget effect.

```
NIN Alliance Trial 1989
  variety !A
  id
  :
  :
  row 22
  column 11
nin89aug.asd !skip 1
yield ~ mu variety !r units,
!f mv
1 2 0
11 column AR1 0.3
22 row AR1 0.3
```

4 Two-dimensional separable autoregressive spatial model with random replicate effects

This is essentially a combination of **2b** and **3c** to demonstrate specifying an R structure and a G structure in the same model. The variance header line `1 2 1` indicates that there is one R structure (1) that involves two variance models (2) and is therefore the direct product of two matrices, and there is one G structure (1). The R structures are defined first so the next two lines are the R structure definition lines for `e`, as in **3b**. The last two lines are the G structure definition lines for `repl`, as in **2b**. In this case $\mathbf{V} = \sigma_e^2 (\gamma_r \mathbf{I}_{242} + \boldsymbol{\Sigma}_c(\rho_c) \otimes \boldsymbol{\Sigma}_r(\rho_r))$.

See Section 7.4

```
NIN Alliance Trial 1989
  variety !A
  id
  :
  :
  row 22
  column 11
nin89aug.asd !skip 1
yield ~ mu variety !r repl,
!f mv
1 2 1
11 column AR1 0.3
22 row AR1 0.3
repl 1
repl 0 IDV 0.1
```


5 Two-dimensional separable autoregressive spatial model defined as a G structure

This model is equivalent to **3c** but with the spatial model defined as a G structure rather than an R structure. As discussed in **2b**,

See Section 7.7

- *all but one* of the models supplied when the G structure involves more than one variance model must be correlation models,
- the other model must not be a correlation model, that is, the other model must be either an homogeneous or a heterogeneous variance model, and an initial value for the scale parameter must be supplied.

```
NIN Alliance Trial 1989
  variety !A
  id
  :
  row 22
  column 11
nin89.asd !skip 1
yield ~ mu variety,
!r row.column
0 0 1
row.column 2
row 0 AR1V 0.3 0.1
column 0 AR1 0.3
```

For this reason, the model for rows is now

AR1V and an initial value of 0.1 has been supplied for the scale parameter. In this case $\mathbf{V} = \sigma_e^2 (\gamma_{rc} \boldsymbol{\Sigma}_c(\rho_c) \otimes \boldsymbol{\Sigma}_r(\rho_r) + I_{224})$. Use of `row.column` as a G structure is a useful approach for analysing incomplete spatial arrays. This approach will often run faster for large trials but requires more memory.

Important

Note that we have used the original version of the data and `!f mv` is omitted from this analysis since `row.column` is fitted as a G structure. If we had used the augmented data `nin89aug.asd` we would still omit `!f mv` and ASReml would discard the records with missing yield.

Table 7.1: Sequence of variance structures for the NIN field trial data

	ASReml syntax	extra random terms		residual error term			
		term	G structure models		term	R structure models	
			1	2		1	2
1	<code>yield ~ mu variety repl</code>	-	-	error	ID	-	
2a	<code>yield ~ mu variety, !r repl</code>	repl	IDV	error	ID	-	
2b	<code>yield ~ mu variety, !r repl 0 0 1 repl 1 4 0 IDV 0.1</code>	repl	IDV	error	ID	-	
3a	<code>yield ~ mu variety, !f mv 1 2 0 11 column ID 22 row AR1 0.3</code>	-	-	column.row	ID	AR1	
3b	<code>yield ~ mu variety, !f mv 1 2 0 11 column AR1 0.3 22 row AR1 0.3</code>	-	-	column.row	AR1	AR1	
3c	<code>yield ~ mu variety, !r units !f mv 1 2 0 11 column AR1 0.3 22 row AR1 0.3</code>	units	IDV	column.row	AR1	AR1	
4	<code>yield ~ mu variety, !r repl !f mv 1 2 1 11 column AR1 0.3 22 row AR1 0.3 repl 1 4 0 IDV 0.1</code>	repl	IDV	column.row	AR1	AR1	
5	<code>yield ~ mu variety, !r column.row 0 0 1 column.row 2 column 0 AR1 .3 row 0 AR1V 0.3 0.1</code>	column.row	AR1	AR1V	error	ID	

7.4 Variance structures

The previous sections have introduced variance modelling in ASReml using the NIN data for demonstration. In this and the remaining sections the syntax is described formally. However, where appropriate we continue to reference the example.

General syntax

Variance model specification in ASReml has the following general form

```
[variance header line  
[R structure definition lines ]  
[G structure header and definition lines ]  
[variance parameter constraints ]]
```

- *variance header line* specifies the number of R and G structures,
- *R structure definition lines* define the R structures (variance models for error) as specified in the variance header line,
- *G structure header and definition lines* define the G structures (variance models for the additional random terms in the model) as specified in the variance header line; these lines are always placed after any R structure definition lines,
- *variance parameter constraints* are included if parameter constraints are to be imposed, see the !VCC *c* qualifier in Table 5.5 and Section 7.9 on constraints between and within variance structures.

A schematic outline of the variance model specification lines (variance header line, and R and G structure definition lines) is presented in Table 7.2 using the variance model of **4** for demonstration.

Table 7.2: Schematic outline of variance model specification in ASReml

general syntax			model 4
<i>variance header line</i>	[s [c [g]]]		1 2 1

<i>R structure definition lines</i>	S_1	C_1	11 column AR1 0.3
		C_2	22 row AR1 0.3
		⋮	⋮
		C_c	-
	S_2	C_1	-
		⋮	⋮
		C_c	-
	⋮	⋮	⋮
	S_s	C_1	-
		⋮	⋮
		C_c	-

<i>G structure definition lines</i>	G_1		repl 1 4 0 IDV 0.1
	G_2		-
	⋮		⋮
	G_g		-

Variance header line

The variance header line is of the form

$[s [c [g]]]$

- s and c relate to the R structures, g is the number of G structures,
- the variance header line may be omitted if the default IID R structure is required, no G structures are being explicitly defined and there are no parameter constraints (see !VCC and examples **1** and **2a**),
- s is used to code the number of independent sections in the error term
 - if $s = 0$, the default IID R structure is assumed and no R structure definition lines are required (as in examples **2b** and **5**),
 - if $s > 0$, s R structure definitions are required, one for each of the s sections (as in examples **3a**, **3b**, **3c** and **4**),
 - for the analysis of multi-section data s can be replaced by the name of a factor with the appropriate number of levels, one for each section,
- c is the number of component variance models involved in the variance structure for the error term for each section; for example, **3a**, **3b** and **3c** have `column.row` as the error term and the variance structure for `column.row` involves 2 variance models, the first for `column` and the second for `row`,
 - c has a default value of 2 when s is not specified as zero,
- g is the number of variance structures (G structures) that will be explicitly specified for the random terms in the model.

```
NIN Alliance Trial 1989
variety !A
id
:
:
row 22
column 11
nin89aug.asd !skip 1
yield ~ mu variety !r repl,
!f mv
1 2 1
22 row AR1 0.3
11 column AR1 0.3
repl 1
repl 0 IDV 0.1
```

R and G structures are now discussed with reference to s , c and g . As already noted, each variance structure may involve several variance models which relate to the individual terms involved in the random effect or error. For example, a two factor interaction may have a variance model for each of the two factors involved in the interaction. Variance models are listed in Table 7.3. As indicated in the discussion of **2b**, care must be taken with respect to scale parameters when combining variance models (see also Section 7.7).

See Table 7.3
See Section 7.7

R structure definition

For each of the s sections there must be c R structure definitions. Each definition may take several lines. Each R structure definition specifies a variance model and has the form

```
order [field model [initial_values] [qualifiers]
[additional_initial_values]]
```

- *order* is either the number of levels in the corresponding term or the name of a factor that has the same number of levels as the term, for example,

```
11 column AR1 0.5
```

is equivalent to

```
column column AR1 0.5
```

when `column` is a factor with 11 levels,

- *field* is the name of the data field (variate or factor) that corresponds to the term and therefore indexes the levels of the term;

- ASReml uses this field to sort the units so they match the R structure,
- in the example the data will be sorted internally rows within columns for the analysis but the residuals will be printed in the `.yht` file in the original order (which is actually rows within columns in this case).

Important It is assumed that the joint indexing of the components uniquely defines the experimental units,

- if *field* is a variable, it can be plot coordinates provided the plots are in a regular grid. Thus in this example

```
11 lat AR1 0.3
22 long AR1 0.3
```

is valid because `lat` gives column position and `long` gives row position, and the positions are on a regular grid. The autoregressive correlation values will still be on an plot index basis (1, 2, 3, ...), not on a distance basis (10m, 20m, 30m, ...),

- if the data is sorted appropriately for the order the models are specified, set *field* to 0,

- *model* specifies the variance model for the term, for example,

```
NIN Alliance Trial 1989
variety !A
:
:
row 22
column 11
nin89aug.asd !skip 1
yield ~ mu variety !r repl,
!f mv
1 2 1
11 column AR1 0.3
22 row AR1 0.3
repl 1
repl 0 IDV 0.1
```

```
22 row AR1 0.3
```

chooses a first order autoregressive model for the row error process,

- all the variance models available in ASReml are listed in Table 7.3,
- these models have associated variance parameters,
- a error variance component (σ_e^2 for the example, see Section 7.3) is automatically estimated for each section,
- the default *model* is ID,

- *initial_values* are initial or starting values for the variance parameters and must be supplied, for example,

```
22 row AR1 0.3
```

chooses an autoregressive model for the row error process (see Table 7.1) with a starting value of 0.3 for the row correlation,

- *qualifiers* tell ASReml to modify the variance model in some way; the qualifiers are described in Table 7.4,
- *additional_initial_values* are read from the following lines if there are not enough initial values on the model line. Each variance model has a certain number of parameters. If insufficient non zero values are found on the model line ASReml expects to find them on the following line(s),
 - initial values of 0.0 will be ignored if they are on the model line but are accepted on subsequent lines,
 - the notation *n*v* (for example, 5 * 0.1) is permitted on subsequent lines (but not the model line) when there are *n* repeats of a particular initial value *v*,
 - only in a few specified cases is 0 permitted as an initial value of a non-zero parameter.

G structure header and definition lines

There are g sets of G structure definition lines and each set is of the form

```

model_term d
order [key model [initial_values] [qualifier]
[additional_initial_values]]
order [key model [initial_values] [qualifier]
[additional_initial_values]]
⋮
order [key model [initial_values] [qualifier]
[additional_initial_values]]

```

- *model_term* is the term from the linear model to which the variance structure applies; the variance structure may cover additional terms in the linear model, see Section 7.8
- *d* is the number of variance models and hence direct product matrices involved in the G structure; the following lines define the d variance models,
- *order* is either the number of levels in the term or the name of a factor that has the same number of levels as the component,
- *key* is usually zero but for power models (EXP, GAU,...) provides the distance data needed to construct the model,
- *model* is the ASReml variance model identifier/acronym selected for the term,
 - variance models are listed in Table 7.3,
 - these models have associated variance parameters,
- *initial_values* are initial or starting values for the variance parameters, the values for initial values are as described above for R structure definition lines,
- *qualifier* tells ASReml to modify the variance model in some way; the qualifiers are described in Table 7.4.

```

NIN Alliance Trial 1989
  variety !A
  id
  ⋮
  ⋮
  row 22
  column 11
  nin89aug.asd !skip 1
  yield ~ mu variety !r repl,
  !f mv
  1 2 1
  22 row AR1 0.3
  11 column AR1 0.3
  repl 1
  repl 0 IDV 0.1

```

7.5 Variance model description

Table 7.3 presents the full range of variance models, that is, correlation, homogeneous variance and heterogeneous variance models available in ASReml. The

table contains the model identifier, a brief description, its algebraic form and the number of parameters. The first section defines (BASE) correlation models and in the next section we show how to extend them to form variance models. The second section defines some models parameterized as variance/covariance matrices rather than as correlation matrices. The third section covers some special cases where the covariance structure is known except for the scale.

Forming variance models from correlation models

The base identifiers presented in the first part of Table 7.3 are used to specify the correlation models. The corresponding homogeneous and heterogeneous variance models are specified by appending **V** and **H** to the base identifiers respectively. This convention holds for most models. However, no **V** or **H** should be appended to the base identifiers for the heterogeneous variance models at the end of the table (from **DIAG** on).

In summary, to specify

- a correlation model, provide the base identifier given in Table 7.3, for example

`EXP .1`

is an exponential correlation model,

- an homogeneous variance model, append a **V** to the base identifier and provide an additional initial value for the variance, for example,

`EXPV .1 .3`

is an exponential variance model,

- a heterogeneous variance model, append an **H** to the base identifier and provide additional initial values for the diagonal variances, for example,

`CORUH .1 .3 .4 .2`

is a 3-dimensional uniform correlation variance model, heterogeneous form.

Important See Section 7.7 for rules on combining variance models and important notes regarding initial values.

The algebraic forms of the homogeneous and heterogeneous variance models are determined as follows. Let $\mathbf{C}^{(\omega \times \omega)} = [C_{ij}]$ denote the correlation matrix for a particular correlation model. If $\mathbf{\Sigma}^{(\omega \times \omega)}$ is the corresponding homogeneous variance matrix then

$$\mathbf{\Sigma} = \sigma^2 \mathbf{C}.$$

It has just one more parameter than the correlation model. For example, the homogeneous variance model corresponding to the ID correlation model has variance matrix $\Sigma = \sigma^2 I_\omega$ (specified IDV in the ASReml command file, see below) and one parameter. The initial values for the variance parameters are listed after the initial values for the correlation parameters. For example, in

```
AR1V 0.3 0.5
```

0.3 is the initial spatial correlation parameter and 0.5 is the initial variance parameter value.

Similarly, if $\Sigma_h^{(\omega \times \omega)}$ is the heterogeneous variance matrix corresponding to C , then

$$\Sigma_h = DCD$$

where $D^{(\omega \times \omega)} = \text{diag}(\sigma_i)$. In this case there are an additional ω parameters. For example, the heterogeneous variance model corresponding to ID has variance matrix

$$\Sigma_h = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_\omega^2 \end{bmatrix}$$

(specified IDH in the ASReml command file, see below) and involves the ω parameters $\sigma_1^2 \dots \sigma_\omega^2$.

Additional notes on variance models

These additional notes are to assist in interpreting Table 7.3.

- the IDH and DIAG models fit the same diagonal variance structure,
- the CORGH and US models fit the same completely general variance structure parameterized differently,
- in CHOL k models $\Sigma = LDL'$ where L is lower triangular with ones on the diagonal, D is diagonal and k is the number of non-zero off diagonals in L ,
- in ANTE k models $\Sigma^{-1} = UDU'$ where U is upper triangular with ones on the diagonal, D is diagonal and k is the number of non-zero off diagonals in U ,
- the CHOL k and ANTE k models are equivalent to the US structure, that is, the full variance structure, when k is $\omega - 1$,

- initial values for **US**, **CHOL** and **ANTE** structures are given in the form of a **US** matrix which is specified lower triangle row-wise, viz

$$\begin{bmatrix} \sigma_{11} & & \\ \sigma_{21} & \sigma_{22} & \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix},$$

that is, initial values are given in the order, $1 = \sigma_{11}$, $2 = \sigma_{21}$, $3 = \sigma_{22}$, ...

- the **US** model is associated with several special features of **ASReml**. When used in the **R** structure for multivariate data, **ASReml** automatically recognises patterns of missing values in the responses (see Chapter 8). Also, there is an option to update its values by **EM** rather than **AI** when its **AI** updates make the matrix non positive definite.
- power models rely on the definition of distance for the associated term, for example,
 - the distance between time points in a one-dimensional longitudinal analysis,
 - the spatial distance between plot coordinates in a two-dimensional field trial analysis.

Information for determining distances is supplied by the *key* argument on the structure line.

- For one dimensional cases, *key* may be
 - * the name of a data field containing the coordinate values
 - * 0 in which case a vector of coordinates of length *order* must be supplied after all **R** and **G** structure lines.
- In two directions (**IEXP**, **IGAU**, **IEUC**, **AEXP**, **AGAU**) the *key* argument in the **G** structure definition has either
 - * the form *rrcc* where *rr* is the number of a data field containing the coordinates for the first dimension and *cc* is the number of a data field containing the coordinates for the second direction. For example, in the analysis of spatial data, if the *x* coordinate was in field 3 and the *y* coordinate was in field 4, the second argument would be 304.
 - * or when used with a factor created by the **fac()** model term, the syntax would be as demonstrated in the following example

```

⋮
y ~ mu ...!r fac(x,y) ...
⋮
fac(x,y) 1
fac(x,y) fac(x,y) EUCV .7 1.3

```

- **FAk**, **FACV k** and **XFAk** are different parameterizations of the factor analytic model in which Σ is modelled as $\Sigma = \Gamma\Gamma' + \Psi$ where $\Gamma^{(\omega \times k)}$ is a matrix of loadings on the covariance scale and Ψ is a diagonal vector of specific variances. See Smith *et al.* (2001) and Thompson *et al.* (2003) for examples of factor analytic models in multi-environment trials. The general limitations are
 - that Ψ may not include zeros except in the **XFAk** formulation
 - constraints are required in Γ for $k > 1$ for identifiability. Typically, one zero is placed in the second column, two zeros in the third column, etc.
 - The total number of parameters fitted $(k\omega + \omega - k(k-1)/2)$ may not exceed $\omega(\omega + 1)/2$.
- in **FAk** models the variance-covariance matrix $\Sigma^{(\omega \times \omega)}$ is modelled on the correlation scale as $\Sigma = DCD$, where
 - $D^{(\omega \times \omega)}$ is diagonal,
 - $DD = \text{diag}(\Sigma)$,
 - $C^{(\omega \times \omega)}$ is a correlation matrix of the form $FF' + E$ where $F^{(\omega \times k)}$ is a matrix of loadings on the correlation scale and E is diagonal and is defined by difference so that $(FF' + E)$ is a correlation matrix (diagonal elements equal 1),
 - the parameters are specified in the order *loadings for each factor (F) followed by the variances* ($\text{diag}(\Sigma)$); when k is greater than 1, constraints on the elements of F are required, see Table 7.5,
- **FACV k** models (**CV** for *covariance*) are an alternative formulation of **FA** models in which Σ is modelled as $\Sigma = \Gamma\Gamma' + \Psi$ where $\Gamma^{(\omega \times k)}$ is a matrix of loadings on the covariance scale and Ψ is diagonal
 - the parameters are specified in the order *loadings (Γ) followed by variances (Ψ)*; when k is greater than 1, constraints on the elements of Γ are required, see Table 7.5,
 - the parameters in **FACV** are related to those in **FA** by $\Gamma = DF$ and $\Psi = DED$,
- **XFAk** (**X** for *extended*) is the third form of the factor analytic model and has the same parameterisation as for **FACV**, that is, $\Sigma = \Gamma\Gamma' + \Psi$. However, **XFA** models
 - the parameters are specified in the order $\text{diag}(\Psi)$ and $\text{vec}(\Gamma)$; when k is greater than 1, constraints on the elements of Γ are required, see Table 7.5,
 - can only be used in **G** structures in combination with the `xfa(f,k)` model term,

- return the factors as well as the effects.
- permit some elements of Ψ to be fixed to zero,
- may not be used in R structures,
- are computationally faster than the FACV formulation for large problems when k is much smaller than ω ,

Special consideration is required when using the $XFAk$ model. The SSP must be expanded to have room to hold the k factors. This is achieved by using the $xfa(f, k)$ model term in place of f in the model. For example,

```
y ~ site !r geno.xfa(site,2)
0 0 1
geno.xfa(site,2) 2
geno
xfa(site,2) 0 XFA2
```

- the OWN variance structure is a facility whereby users may specify their own variance structure. This facility requires the user to supply a program MYOWNGDG that reads the current set of parameters, forms the G matrix and a full set of derivative matrices, and writes these to disk. Before each iteration, ASReml writes the OWN parameters to a file, runs MYOWNGDG (which it presumes forms the G and derivative matrix) and then reads the matrices back in. An example of MYOWNGDG.f90 is distributed with ASReml. It duplicates the AR1 and AR2 structures. The following job fits an AR2 structure using this program.

Example of using the OWN structure

```
rep
blcol
blrow
variety 25
yield
shf.asd !skip 1
y ~ va
1 2
10 0 AR1 .1
15 0 OWN2 .2 .1 !TRR
```

The file written by ASReml has extension `.own` and looks like

```
15 2 1
0.6025860D+000.1164403D+00
This file was written by asreml for reading by your
program MYOWNGDG
```

asreml writes this file, runs your program and then reads shfown.gdg which it presumes has the following format: The first lines should agree with the top of this file specifying the order of the matrices (15) the number of variance parameters (2) and a control parameter you can specify (1). These are written in (3I5) format. They are followed by the list of variance parameters written in (6D13.7) format. Follow this with 3 matrices written in (6D13.7) format. These are to be each of 120 elements being lower triangle row-wise of the G matrix and its derivatives with respect to the parameters in turn.

This file contains details about what is expected in the file written by your program. The filename used has the same basename as the job you are running with extension .own for the file written by ASReml and .gdg for the file your program writes. The type of the parameters is set with the !T qualifier described below. The control parameter is set using the !F qualifier.

- !F2 applies to OWN models. With OWN, the argument of !F is passed to the MYOWNGDG program as an argument the program can access. This is the mechanism that allows several OWN models to be fitted in a single run.
- !Ts is used to set the type of the parameters. It is primarily used in conjunction with the OWN structure as ASReml knows the type in other cases. The valid type codes are as follows:

code	description	action if !GP is set
V	variance	forced positive
G	variance ratio	forced positive
R	correlation	$-1 < r < 1$
C	covariance	
P	positive correlation	$0 < r < 1$
L	loading	

This coding also affects whether the parameter is scaled by σ^2 in the output report.

Table 7.3: Details of the variance models available in ASReml

base identifier	description	algebraic form	number of parameters [†]		
			corr	homo's variance	hetero's variance
Correlation models					
ID	identity	$C_{ii} = 1, C_{ij} = 0, i \neq j$	0	1	ω
AR[1]	1 st order autoregressive	$C_{ii} = 1, C_{i+1,i} = \phi_1$ $C_{ij} = \phi_1 C_{i-1,j}, i > j + 1$ $ \phi_1 < 1$	1	2	$1 + \omega$
SAR	symmetric autoregressive	$C_{ii} = 1,$ $C_{i+1,i} = \phi_1 / (1 + \phi_1^2 / 4)$ $C_{ij} = \phi_1 C_{i-1,j} - \phi_1^2 / 4 C_{i-2,j},$ $i > j + 1$ $ \phi_1 < 1$	1	2	$1 + \omega$
AR2	2 nd order autoregressive	$C_{ii} = 1,$ $C_{i+1,i} = \phi_1 / (1 - \phi_2)$ $C_{ij} = \phi_1 C_{i-1,j} + \phi_2 C_{i-2,j}, i > j + 1$ $ \phi_1 < (1 - \phi_2), \phi_2 < 1$	2	3	$2 + \omega$
MA[1]	1 st order moving average	$C_{ii} = 1,$ $C_{i+1,i} = -\theta_1 / (1 + \theta_1^2)$ $C_{ji} = 0, j > i + 2$ $ \theta_1 < 1$	1	2	$1 + \omega$
MA2	2 nd order moving average	$C_{ii} = 1,$ $C_{i+1,i} = -\theta_1(1 - \theta_2) / (1 + \theta_1^2 + \theta_2^2)$ $C_{i+2,i} = -\theta_2 / (1 + \theta_1^2 + \theta_2^2)$ $C_{ji} = 0, j > i + 2$ $\theta_2 \pm \theta_1 < 1$ $ \theta_1 < 1, \theta_2 < 1$	2	3	$2 + \omega$
ARMA	autoregressive moving average	$C_{ii} = 1,$ $C_{i+1,i} = (\theta - \phi)(1 - \theta\phi) / (1 + \theta^2 - 2\theta\phi)$ $C_{ji} = \phi C_{j-1,i}, j > i + 1$ $ \theta < 1, \phi < 1$	2	3	$2 + \omega$
CORU	uniform correlation	$C_{ii} = 1, C_{ij} = \phi, i \neq j$	1	2	$1 + \omega$

Details of the variance models available in ASReml

base identifier	description	algebraic form	number of parameters [†]		
			corr	homo's variance	hetero's variance
CORB	banded correlation	$C_{ii} = 1$ $C_{i+j,i} = \phi_j, 1 \leq j \leq \omega - 1$ $ \phi_j < 1$	$\omega - 1$	ω	$2\omega - 1$
CORG	general correlation CORGH = US	$C_{ii} = 1$ $C_{ij} = \phi_{ij}, i \neq j$ $ \phi_{ij} < 1$	$\frac{\omega(\omega-1)}{2}$	$\frac{\omega(\omega-1)}{2} + 1$	$\frac{\omega(\omega-1)}{2} + \omega$
One-dimensional power models					
EXP	exponential	$C_{ii} = 1$ $C_{ij} = \phi^{ x_i - x_j }, i \neq j$ x_i are <i>coordinates</i> $0 < \phi < 1$	1	2	$1 + \omega$
GAU	gaussian	$C_{ii} = 1$ $C_{ij} = \phi^{(x_i - x_j)^2}$ x_i are <i>coordinates</i> $0 < \phi < 1$	1	2	$1 + \omega$
Two-dimensional power models					
IEXP	isotropic exponential	$C_{ii} = 1$ $C_{ij} = \phi^{ x_i - x_j + y_i - y_j }$ \mathbf{x} and \mathbf{y} <i>vectors of coordinates</i> $0 < \phi < 1$	1	2	$1 + \omega$
IGAU	isotropic gaussian	$C_{ii} = 1$ $C_{ij} = \phi^{(x_i - x_j)^2 + (y_i - y_j)^2}$ \mathbf{x} and \mathbf{y} <i>vectors of coordinates</i> $0 < \phi < 1$	1	2	$1 + \omega$
IEUC	isotropic euclidean	$C_{ii} = 1$ $C_{ij} = \phi^{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}$ \mathbf{x} and \mathbf{y} <i>vectors of coordinates</i> $0 < \phi < 1$	1	2	$1 + \omega$

Details of the variance models available in ASReml

base identifier	description	algebraic form	number of parameters [†]		
			corr	homo's variance	hetero's variance
AEXP	anisotropic exponential	$C_{ii} = 1$ $C_{ij} = \phi_1^{ x_i - x_j } \phi_2^{ y_i - y_j }$ <i>\mathbf{x} and \mathbf{y} vectors of coordinates</i> $0 < \phi_1 < 1, 0 < \phi_2 < 1$	2	3	$2 + \omega$
AGAU	anisotropic gaussian	$C_{ii} = 1$ $C_{ij} = \phi_1^{(x_i - x_j)^2} \phi_2^{(y_i - y_j)^2}$ <i>\mathbf{x} and \mathbf{y} vectors of coordinates</i> $0 < \phi_1 < 1, 0 < \phi_2 < 1$	2	3	$2 + \omega$
Additional heterogeneous variance models					
DIAG	diagonal = IDH	$\Sigma_{ii} = \phi_i, \Sigma_{ij} = 0, i \neq j$	-	-	ω
US	unstructured general covariance matrix	$\Sigma_{ij} = \phi_{ij}$	-	-	$\frac{\omega(\omega+1)}{2}$
OWN k	user explicitly forms \mathbf{V} and $\partial\mathbf{V}$		-	-	k
ANTE[1] ANTE k	1^{st} k order k^{th} antedependence $1 \leq k \leq \omega - 1$	$\Sigma^{-1} = \mathbf{U}\mathbf{D}\mathbf{U}'$ $D_{ii} = d_i, D_{ij} = 0, i \neq j$ $U_{ii} = 1, U_{ij} = u_{ij}, 1 \leq j - i \leq k$ $U_{ij} = 0, i > j$	-	-	$\frac{\omega(\omega+1)}{2}$
CHOL[1] CHOL k	1^{st} k order k^{th} cholesky $1 \leq k \leq \omega - 1$	$\Sigma = \mathbf{L}\mathbf{D}\mathbf{L}'$ $D_{ii} = d_i, D_{ij} = 0, i \neq j$ $L_{ii} = 1, L_{ij} = l_{ij}, 1 \leq i - j \leq k$	-	-	$\frac{\omega(\omega+1)}{2}$
FA[1] FA k	1^{st} k order k^{th} factor analytic	$\Sigma = \mathbf{D}\mathbf{C}\mathbf{D}$, $\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{E}$, <i>\mathbf{F} contains k correlation factors</i> <i>\mathbf{E} diagonal</i> $\mathbf{D}\mathbf{D} = \text{diag}(\Sigma)$	-	-	$\omega + \omega$ $k\omega + \omega$

Details of the variance models available in ASReml

base identifier	description	algebraic form	number of parameters [†]		
			corr	homo's variance	hetero's variance
FACV[1] FACV k	1 st k th order factor analytic covariance form	$\Sigma = \Gamma\Gamma' + \Psi$, Γ contains covariance factors Ψ contains specific variance	-	-	$\omega + \omega$ $k\omega + \omega$
XFA[1] XFA k	1 st k th extended factor analytic covariance form	$\Sigma = \Gamma\Gamma' + \Psi$, Γ contains covariance factors Ψ contains specific variance	-	-	$\omega + \omega$ $k\omega + \omega$
Inverse relationship matrices[‡]					
AINV	inverse relationship matrix derived from pedigree		0	1	-
GIV1	generalised inverse number 1		0	1	-
⋮	⋮	⋮	⋮	⋮	
GIV6	generalised inverse number 6		0	1	-

[†] This is the number of values the user must supply as initial values where ω is the dimension of the matrix. The homogeneous variance form is specified by appending V to the correlation basename; the heterogeneous variance form is specified by appending H to the correlation basename

[‡] These must be associated with 1 variance parameter unless used in direct product with another structure which provides the variance.

7.6 Variance structure qualifiers

Table 7.4 describes the R and G structure line qualifiers.

Table 7.4: List of R and G structure qualifiers

<i>qualifier</i>	action
<code>!=s</code>	used to constrain parameters within variance structures, see Section 7.9.
<code>!GP, !GU, !GF, !GZ</code>	<p>modify the updating of the variance parameters. The exact action of these codes in setting bounds for parameters depends on the particular model.</p> <p><code>!GP</code> (the default in most cases) attempts to keep the parameter in the theoretical parameter space and is activated when the update of a parameter would take it outside its space. For example, if an update would make a variance negative, the negative value is replaced by a small positive value. Under the <code>!GP</code> condition, repeated attempts to make a variance negative are detected and the value is then <i>fixed</i> at a small positive value. This is shown in the output in that the parameter will have the code B rather than P appended to the value in the variance component table.</p> <p><code>!GU</code> (unrestricted) does not limit the updates to the parameter. This allows variance parameters to go negative and correlation parameters to exceed ± 1. Negative variance components may lead to problems; the mixed model coefficient matrix may become non-positive definite. In this case the sequence of REML log-likelihoods may be erratic and you may need to experiment with starting values.</p> <p><code>!GF</code> fixes the parameter at its starting value</p> <p><code>!GZ</code> only applies to FA and FACV models and fixes the parameter in an FA factor to ZERO</p> <p>For multiple parameters, the form <code>!GXXXX</code> can be used to specify F, P, U or Z for the parameters individually. A shorthand notation allows a repeat count before a code letter. Thus <code>!GPPPPPPPPPPPPPPZPPPZP</code> could be written as <code>!G14PZ3PZP</code>.</p> <p>For a US model, <code>!GP</code> forces ASReml to attempt to keep the matrix positive definite. After each AI update, it extracts the eigenvalues of the updated matrix. If any are negative or zero, the AI update is discarded and an EM update is performed. Notice that the EM update is applied to all of the variance parameters in the particular US model and cannot be applied to only a subset of them.</p>

List of R and G variance structure definition line qualifiers

<i>qualifier</i>	action
<code>!S2=value</code>	sets the value of the error variance within the section: the <code>!S2=</code> qualifier may be used in R structure definitions to represent the residual variance associated with the particular section. There is always an error variance parameter associated with a section R structure. In multiple section analyses the <code>!S2=</code> qualifier is used on the first of the <i>c</i> lines for each section to set the initial error variance for that section. If this is not supplied, ASReml calculates an initial value that is half the simple variance of the data in the site. To fix the variance <code>!S2=v</code> is used.
<code>!S2==value</code>	is used in a similar manner to <code>!S2=</code> except that the variance is fixed at the <i>value</i> .
<code>!S2==1</code>	is often used in the analysis of multiple section data and in multivariate analyses and when variance parameters are included in the R structure models.

7.7 Rules for combining variance models

As noted in Section 2.1 under Combining variance models, variance structures are sometimes formed by combining variance models. For example, a two factor interaction may involve two variance models, one for each of the two factors in the interaction. Some of the rules for combining variance models differ for R structures and G structures. A summary of the rules is as follows:

- when combining variance models in both R and G structures, the resulting direct product structure must match the ordered effects with the outer factor first, for example, the G structure in the example opposite is for `column.row` which tells ASReml that the direct product structure matches the effects ordered rows *within* columns. This is why the G structure definition line for `column` is specified first,
- ASReml automatically includes and estimates an error variance parameter for each section of an R structure. The variance structures defined by the user should therefore normally be correlation matrices. A variance model can be specified

```
NIN Alliance Trial 1989
variety !A
id
:
row 22
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl,
!f mv
0 0 1
column.row 2
column 0 AR1 0.4
row 0 ARV1 0.3 0.1
```

but the `!S2==1` qualifier would then be required to fix the error variance at 1 and prevent ASReml trying to estimate two confounded parameters (error variance and the parameter corresponding to the variance model specified, see **3a** on page 83),

See Sections 2.1
and 7.5

- ASReml does not have an implicit scale parameter for G structures that are defined explicitly. For this reason the model supplied when the G structure involves just one variance model must be a variance model; an initial value must be supplied for this associated scale parameter; this is discussed under *additional_initial_values* on page 91,
- when the G structure involves more than one variance model, one must be either a homogeneous or a heterogeneous variance model and the rest should be correlation models; if more than one are non-correlation models then the `!GF` qualifier should be used to avoid identifiability problems, that is, ASReml trying to estimate confounded parameters.

7.8 G structures involving more than one random term

The usual case is that a variance structure applies to a particular term in the linear model and that there is no covariance between model terms. Sometimes it is appropriate to include a covariance. Then, it is essential that the model terms be listed together and that the variance structure defined for the first term be the structure required for both terms. When the terms are of different size, the terms should be linked together with the `!{` and `!}` qualifiers (Table 6.1). While ASReml will check the overall size, it does not check that the order of effects matches the structure definition so the user must be careful to get this right. Check that the terms are conformable by considering the order of the fitted effects and ensuring the first term of the direct product corresponds to the outer factor in the nesting of the effects. Two examples are

- **random regressions** where we want a covariance between intercept and slope

```

:
!r animal animal.time
:
animal 2
2 0 US 3 -.5 2
animal

```

is equivalent (though not identical because of the scaling changes) to

```

:

```

```

!r pol(time,1).animal
:
pol(time,1).animal 2
pol(time,1) 0 US 1 -.1 .2
animal

```

- **maternal/direct genetic covariance**

```

lambid !P
sireid !P
damid !P
:
wwt ywt ~ Trait Trait.sex !r !{ Trait.lambid at(Trait,2).damid !}
:
Trait.lambid 2
3 0 US
1.3          # Var(wwt_D)
1.0 2.2      # Cov(wwt_D,ywt_D) Var(ywt_D)
-.1 -.2 0.8  # Cov(wwt_D,wwt_M) Cov(ywt_D,wwt_M) Var(wwt_M)
lambid 0 AINV # AINV explicitly requests to use A inverse

```

7.9 Constraining variance parameters

Parameter constraints within a variance model

Equality of parameters in a variance model can be specified using the `!=s` qualifier where `s` is a string of letters and/or zeros (see Table 7.4). Positions in the string correspond to the parameters of the variance model:

- all parameters with the same letter in the structure are treated as the same parameter,
- A to Z matches a to z, so that 26 equalities at most can be formed,
- all parameters corresponding to 0 in the string are unconstrained.

Examples are presented in Table 7.5.

Table 7.5: Examples of constraining variance parameters in ASReml

ASReml code	action
<code>!=ABACBAOCBA</code>	constrain all parameters corresponding to A to be equal, similarly for B and C . The 7th parameter would be left unconstrained. This sequence applied to an unstructured 4×4 matrix would make it banded, that is A B A C B A 0 C B A
<code>site.gen 2 # G header line site 0 US .3 !=OA0AAO !GUPUUP .1 .4 .1 .1 .3 gen</code>	this example defines a structure for the genotype by site interaction effects in a MET in which the genotypes are independent random effects within sites but are correlated across sites with equal covariance.
<code>site 0 FA2 !G4PZ3P4P !=0000000VVVV 4*.9 # initial values for 1st factor 0 3*.1 # initial values for 2nd factor # first fixed at 0 4*.2 # init values for site variances</code>	a 2 factor Factor Analytic model for 4 sites with equal variance is specified using this syntax. The first loading in the second factor is constrained equal to 0 for identifiability. P places restrictions on the magnitude of the loadings and the variances to be positive.
<code>xfa(site,2) 0 XFA2 !=VVVV0 !4P4PZ3P 4*.2 # initial specific variances 4*1.2 # initial loadings for 1st factor 0 3*.3 # initial loadings for 2nd factor</code>	a 2 factor Factor analytic model in which the specific variances are all equal.

Constraints between and within variance models

More general relationships between variance parameters can be defined using the `!VCCc` qualifier placed on the data file definition line.

- `!VCC c` specifies that there are c constraint lines defining constraints to be applied,
- the constraint lines occur after the variance header line and any R and G structure lines, that is, there must be a variance header line,
- each constraint is specified in a separate line in the form

$$P_1 * V_1 \quad P_2 * V_2 \quad \dots$$

- P_i is the number of a parameter and V_i is a coefficient,
 - P_1 is the primary parameter number,
 - * indicates that the next values (V_i) is a weighting coefficient,
 - if the coefficient is 1 you may omit the * 1,
 - if the coefficient is -1 you may write $-P_i$ instead of $P_i * -1$,
 - the meaning of the coefficients is as follows $P_2 = V_2 * P_1 / V_1$,
 - typically $V_1 = 1$,
 - a variance parameter may only be included in constraints once,
- the P_i terms refer to positions in the full variance parameter vector. This may change if the model is changed. The full vector includes a term for each factor in the model and then a term for each parameter defined in the R and G structures. A list of P_i numbers and their initial values is returned in the `.asl` file to help you to check the numbers. Alternatively, examine the `.asr` file from an initial run with `!VCC` included but no arguments supplied. The job will terminate but ASReml will provide the P_i values associated with each variance component. Otherwise the numbers are given in the `.res` file.

The following are examples:

ASReml code	action
<code>5 7 * .1</code>	parameter 7 is a tenth of parameter 5
<code>5 -7</code>	parameter 7 is the negative of parameter 5
<code>32 34 35 37 38 39</code>	for a (4×4) US matrix given by parameters 31 ... 40, the covariances are forced to be equal.

7.10 Model building using the !CONTINUE qualifier

In complex models, the Average Information algorithm can have difficulty maximising the REML log-likelihood when starting values are not reasonably close. ASReml has several internal strategies to cope with this problem but these are not always successful.

When the user needs to provide better starting values, one method is to fit a simpler variance model. For example, it can be difficult to guess reasonable starting values for an unstructured variance matrix. A first step might be to assume independence and just estimate the variances. If all the variances are not positive, there is little point proceeding to try and estimate the covariances.

The !CONTINUE qualifier instructs ASReml to retrieve variance parameters from the .rsv file if it exists rather than using the values in the .asr file. When reading the .rsv file, it will take results from some matrices as supplying starting values for other matrices. The transitions recognised are

```
DIAG to CORUH
DIAG to FA1
CORUH to FA1
FA i to FA i+1
FA i to CORGH
FA i to US
CORGH to US
```

The use of the .rsv file with !CONTINUE in this way reduces the need for the user to type in the updated starting values.

The various models may be written in various !PART *s* of the job and controlled by the !DOPART qualifier. When used with the -r qualifier on the command line (see Chapter 12), the output from the various parts has the partnumber appended to the filename. In this case it would be necessary to copy the .rsv files with the new part number before running the next part to take advantage of this facility.

8

Command file: Multivariate analysis

Introduction

Wether trial data

Model specification

Variance structures

Specifying multivariate variance structures in ASReml

The output for a multivariate analysis

More than two response variates

8.1 Introduction

Multivariate analysis is used here in the narrow sense of a multivariate mixed model. There are many other multivariate analysis techniques which are not covered by ASReml. Multivariate analysis is used when we are interested in estimating the correlations between distinct traits (for example, fleece weight and fibre diameter in sheep) and for repeated measures of a single trait.

Wether trial data

Three important traits for the Australian wool industry are the weight of wool grown per year, the cleanness and the diameter of that wool. Much of the wool is produced from wethers and most major producers have traditionally used a particular strain or *bloodline*. In an effort to assess the importance of bloodline differences, many wether trials have now been conducted. The trial analysed here was conducted from 1984 to 1988. It involves 35 teams of wethers representing 27 bloodlines. The teams were run together for three years at Borenore near Orange. The file contains greasy fleece weight (kg), yield (percentage of clean fleece weight to greasy fleece weight) and fibre diameter (microns).

The command file `wether.as` for a basic bivariate analysis of this data is presented to the right. Below is an extract of the first few lines of the data file `wether.dat`.

```
Orange Wether Trial 1984-8
SheepID !I
TRIAL
BloodLine !I
TEAM *
YEAR *
GFW YLD FDIAM
wether.dat !skip 1
GFW FDIAM ~ Trait Trait.YEAR,
!r Trait.TEAM Trait.SheepID
1 2 2
1485 0 ID
Trait 0 US .2 .2 .4
Trait.TEAM 2
Trait 0 US
0.4
0.3 1.3
TEAM 0 ID
Trait.SheepID 2
Trait 0 US
0.2 0.2 2
SheepID 0 ID
predict YEAR Trait
```

```

SheepID Site Bloodline Team Year GFW Yield FD
0101 3 21 1 1 5.6 74.3 18.5
0101 3 21 1 2 6.0 71.2 19.6
0101 3 21 1 3 8.0 75.7 21.5
0102 3 21 1 1 5.3 70.9 20.8
0102 3 21 1 2 5.7 66.1 20.9
0102 3 21 1 3 6.8 70.3 22.1
0103 3 21 1 1 5.0 80.7 18.9
0103 3 21 1 2 5.5 75.5 19.9
0103 3 21 1 3 7.0 76.6 21.9
:
4013 3 43 35 1 7.9 75.9 22.6
4013 3 43 35 2 7.8 70.3 23.9
4013 3 43 35 3 9.0 76.2 25.4
4014 3 43 35 1 8.3 66.5 22.2
4014 3 43 35 2 7.8 63.9 23.3
4014 3 43 35 3 9.9 69.8 25.5
4015 3 43 35 1 6.9 75.1 20.0
4015 3 43 35 2 7.6 71.2 20.3
4015 3 43 35 3 8.5 78.1 21.7

```

8.2 Model specification

The syntax for specifying a multivariate linear model in ASReml is

$$Y\text{-variates} \sim \text{fixed} [\text{!r random}] [\text{!f sparse_fixed}]$$

- *Y-variates* is a list of traits,
- *fixed*, *random* and *sparse_fixed* are as in the univariate case (see Chapter 6) but involve the special term **Trait** and interactions with **Trait**.

The design matrix for **Trait** is has a level (column) for each trait.

- **Trait** by itself fits the mean for each variate,
- In an interaction **Trait.Fac** fits the factor **Fac** for each variate and **Trait.Cov** fits the covariate **Cov** for each variate.

ASReml internally rearranges the data so that n data records containing t traits each becomes n sets of t analysis records indexed by the internal factor **Trait** *i.e.* nt analysis records ordered **Trait** within data record. If the data is already in this long form, use the **!ASMV t** qualifier to indicate that a multivariate analysis is required.

8.3 Variance structures

Using the notation of Chapter 7, consider a multivariate analysis with t traits and n units in which the data are ordered *traits* within *units*. An algebraic expression for the variance matrix in this case is

$$I_n \otimes \Sigma$$

where $\Sigma^{(t \times t)}$ is an unstructured variance matrix.

Specifying multivariate variance structures in ASReml

A more sophisticated error structure is required for multivariate analysis. For a standard multivariate analysis

- the error structure for the residual must be specified as two-dimensional with independent records and an unstructured variance matrix across traits; records may have observations missing in different patterns and these are handled internally during analysis,
- the R structure *must* be ordered traits within units, that is, the R structure definition line for units must be specified before the line for Trait,
- variance parameters are variances not variance ratios,
- the R structure definition line for units, that is, 1485 0 ID, could be replaced by 0 or 0 0 ID; this tells ASReml to fill in the number of units and is a useful option when the exact number of units in the data is not known to the user,
- the error variance matrix is specified by the model Trait 0 US
 - the initial values are for the lower triangle of the (symmetric) matrix specified row-wise,
 - finding reasonable initial values can be a problem. If initial values are written on the next line in the form $q * 0$ where q is $t(t+1)/2$ and t is the number of traits, ASReml will take half of the phenotypic variance matrix of the data as an initial value, see .as file in code box for example,

```
Orange Wether Trial 1984-8
SheepID !I
TRIAL
BloodLine !I
TEAM *
YEAR *
GFW YLD FDIAM
wether.dat !skip 1
GFW FDIAM ~ Trait Trait.YEAR,
!r Trait.TEAM Trait.SheepID
1 2 2
1485 0 ID
Trait 0 US
3*0
Trait.TEAM 2
Trait 0 US
3*0
TEAM 0 ID
Trait.SheepID 2
Trait 0 US
3*0
SheepID 0 ID
predict YEAR Trait
```

- the special qualifiers relating to multivariate analysis are !ASUV and !ASMV t , see Table 5.4 for detail
 - to use an error structure other than US for the residual stratum you must also specify !ASUV (see Table 5.4) and include mv in the model if there are missing values,
 - to perform a multivariate analysis when the data have already been expanded use !ASMV t (see Table 5.4)
 - t is the number of traits that ASReml should expect,
 - the data file must have t records for each multivariate record although some may be coded missing.

8.4 The output for a multivariate analysis

Below is the output returned in the .asr file for this analysis.

```
ASReml 1.0 [31 Aug 2002] Orange Wether Trial 1984-88
16 Sep 2002 16:32:50.370 64.00 Mbyte MSWindow wether
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support *
* Use !{ and !} when fitting covariance across terms *
***** ARG *
Folder: C:\asr\ex\manex
BloodLine !I
QUALIFIERS: !SKIP 1
Reading wether.dat FREE FORMAT skipping 1 lines
traits Bivariate analysis of GFW and FDIAM
Using 1485 records of 1485 read
Model term      Size Type    COL  Minimum   Mean      Maximum  #zero #miss
1 SheepID       521 Factor    1      1    261.0956    521      0     0
2 TRIAL          2          2  3.000     3.000     3.000     0     0
3 BloodLine     27 Factor    3      1    13.4323     27      0     0
4 TEAM          35 Factor    4      1    18.0067     35      0     0
5 YEAR          3 Factor    5      1     2.0391      3      0     0
6 GFW           1 Variate   6  4.100     7.478    11.20     0     0
7 YLD           7          7  60.30    75.11    88.60     0     0
8 FDIAM         1 Variate   8  15.90    22.29    30.60     0     0
9 Trait         2 Traits/Variat
10 Trait.YEAR   6 Interaction 9 Trait   : 2 5 YEAR   : 3
11 Trait.TEAM   70 Interaction 9 Trait   : 2 4 TEAM    : 35
```

```

12 Trait.SheepID1042 Interaction 9 Trait      : 2 1 SheepID      : 521
1485 identity
  2 UnStructure 0.2000 0.2000 0.4000
2970 records assumed sorted 2 within 1485
  2 UnStructure 0.4000 0.3000 1.3000
35 identity
Structure for Trait.TEAM has 70 levels defined
  2 UnStructure 0.2000 0.2000 2.0000
521 identity
Structure for Trait.SheepID has 1042 levels defined
Forming 1120 equations: 8 dense.
Initial updates will be shrunk by factor 0.316
NOTICE: 2 (more) singularities,
convergence 1 LogL=-886.521 S2= 1.0000 2964 df
            2 LogL=-818.508 S2= 1.0000 2964 df
            3 LogL=-755.911 S2= 1.0000 2964 df
            4 LogL=-725.374 S2= 1.0000 2964 df
            5 LogL=-723.475 S2= 1.0000 2964 df
            6 LogL=-723.462 S2= 1.0000 2964 df
            7 LogL=-723.462 S2= 1.0000 2964 df
            8 LogL=-723.462 S2= 1.0000 2964 df
Source      Model terms  Gamma  Component  Comp/SE  % C
final      Residual    UnStruct 1 0.198351 0.198351 21.94 0 U
components Residual    UnStruct 1 0.128890 0.128890 12.40 0 U
           Residual    UnStruct 2 0.440601 0.440601 21.93 0 U
           Trait.TEAM  UnStruct 1 0.374493 0.374493 3.89 0 U
           Trait.TEAM  UnStruct 1 0.388740 0.388740 2.60 0 U
           Trait.TEAM  UnStruct 2 1.36533 1.36533 3.74 0 U
           Trait.SheepID UnStruct 1 0.257159 0.257159 12.09 0 U
           Trait.SheepID UnStruct 1 0.219557 0.219557 5.55 0 U
           Trait.SheepID UnStruct 2 1.92082 1.92082 14.35 0 U
Residual   Covariance/Variance/Correlation Matrix UnStructured
0.4360 is the 0.1984 0.4360
correlation 0.1289 0.4406
Trait.TEAM  Covariance/Variance/Correlation Matrix UnStructured
0.3745 0.5436
0.3887 1.365
Trait.SheepID Covariance/Variance/Correlation Matrix UnStructured
0.2572 0.3124
0.2196 1.921

```

```

Analysis of Variance                DF    F-incr
   9 Trait                          2    5936.27
  10 Trait.YEAR                      4    1096.41
NOTICE: ASREml cannot yet determine the correct denominator DFs for
        these F statistics. They will not exceed the Residual DF
        which applies only if all components are KNOWN.

                Estimate      Standard Error    T-value    T-prev
10 Trait.YEAR
   2  -0.102262      0.290190E-01    -3.52
   3   1.06636      0.290831E-01    36.67    42.07
   5   1.17407      0.433905E-01    27.06
   6   2.53439      0.434880E-01    58.28    32.85

   9 Trait
   7   7.13717      0.107933         66.13
   8  21.0569      0.209095        100.71    78.16

11 Trait.TEAM                        70 effects fitted
12 Trait.SheepID                    1042 effects fitted
SLOPES FOR LOG(ABS(RES)) on LOG(PV) for Section  1
   1.00  1.54
17 possible outliers: see .res file
Finished: 16 Sep 2002 16:32:54.710  LogL Converged

```

8.5 More than two response variates

Wolfinger (1996) summarises a range of variance structures that can be fitted to repeated measures data and demonstrates the models using five weights taken weekly on 27 rats subjected to 3 treatments. The command file for this analysis is presented to demonstrate a multivariate analysis involving more than two response variables (five in this case). Note that the two dimensional structure for common error meets the requirement of independent units and is correctly ordered traits with units. Note also the use of $15 * 0$ for the error variance matrix ($15 = 5(5+1)/2$, see Section 8.3).

```

Wolfinger rat data
  treat !A
  wt0 wt1 wt2 wt3 wt4
rat.dat
wt0 wt1 wt2 wt3 wt4 ~ Trait,
treat Trait.treat
1 2 0
27 0 ID #error variance
Trait 0 US
15 * 0

```


9 Command file: Genetic analysis

Introduction

The command file

The pedigree file

Reading in the pedigree file

Genetic groups

GIV files

The example

Multivariate genetic analysis

9.1 Introduction

In an ‘animal model’ or ‘sire model’ genetic analysis we have data on a set of animals that are genetically linked via a pedigree. The genetic effects are therefore correlated and, assuming normal modes of inheritance, the correlation expected from additive genetic effects can be derived from the pedigree provided all the genetic links are in the pedigree. The additive genetic relationship matrix (sometimes called the numerator relationship matrix) can be calculated from the pedigree. It is actually the *inverse* relationship matrix that is formed by ASReml for analysis.

For the more general situation where the pedigree based inverse relationship matrix is not the appropriate/required matrix, the user can provide a particular general inverse variance (GIV) matrix explicitly in a `.giv` file.

In this chapter we consider data presented in Harvey (1977) using the command file `harvey.as`.

9.2 The command file

In ASReml the `!P` data field qualifier indicates that the corresponding data field has an associated pedigree. The file containing the pedigree (`harvey.ped` in the example) for `animal` is specified immediately prior to the data file definition. See below for the first 20 lines of `harvey.ped` together with the corresponding lines of the data file `harvey.dat`. All individuals appearing in the data file must appear in the pedigree file. When all the pedigree information (individual, male_parent, female_parent) appears as the first three fields of the data file, the data file can double as the pedigree file. In this example the line `harvey.ped !ALPHA` could be replaced with `harvey.dat !ALPHA`. Typically additional individuals providing additional genetic links are present in the pedigree file.

```
Pedigree file example
animal !P
sire !A
dam
lines 2
damage
adailygain
harvey.ped !ALPHA
harvey.dat
adailygain mu lines, !r
animal 0.25
```

9.3 The pedigree file

The pedigree file is used to define the genetic relationships for fitting a genetic animal model and is required if the !P qualifier is associated with a data field. The pedigree file

- has three fields; the identity of an individual (name or tag), its sire and its dam (or maternal grand sire, defined with the !MGS qualifier, Table 9.1); the identity for the individual *must* come first in the file,
- is sorted so that the line giving the pedigree of an individual appears before any line where that individual appears as a parent,
- is read free format; it may be the same file as the data file if the data file is free format and has the necessary identities in the first three fields, see below,
- is specified on the line immediately preceding the data file line in the command file,
- use identity 0 or * for unknown parents.

harvey.ped

```
101 SIRE_1 0
102 SIRE_1 0
103 SIRE_1 0
104 SIRE_1 0
105 SIRE_1 0
106 SIRE_1 0
107 SIRE_1 0
108 SIRE_1 0
109 SIRE_2 0
110 SIRE_2 0
111 SIRE_2 0
112 SIRE_2 0
113 SIRE_2 0
114 SIRE_2 0
115 SIRE_2 0
116 SIRE_2 0
117 SIRE_3 0
118 SIRE_3 0
119 SIRE_3 0
120 SIRE_3 0
:
```

harvey.dat

```
101 SIRE_1 0 1 3 192 390 2241
102 SIRE_1 0 1 3 154 403 2651
103 SIRE_1 0 1 4 185 432 2411
104 SIRE_1 0 1 4 183 457 2251
105 SIRE_1 0 1 5 186 483 2581
106 SIRE_1 0 1 5 177 469 2671
107 SIRE_1 0 1 5 177 428 2711
108 SIRE_1 0 1 5 163 439 2471
109 SIRE_2 0 1 4 188 439 2292
110 SIRE_2 0 1 4 178 407 2262
111 SIRE_2 0 1 5 198 498 1972
112 SIRE_2 0 1 5 193 459 2142
113 SIRE_2 0 1 5 186 459 2442
114 SIRE_2 0 1 5 175 375 2522
115 SIRE_2 0 1 5 171 382 1722
116 SIRE_2 0 1 5 168 417 2752
117 SIRE_3 0 1 3 154 389 2383
118 SIRE_3 0 1 4 184 414 2463
119 SIRE_3 0 1 5 174 483 2293
120 SIRE_3 0 1 5 170 430 2303
:
```

9.4 Reading in the pedigree file

The syntax for specifying a pedigree file in the ASReml command file is

pedigree_file [*qualifiers*]

- the *qualifiers* are listed in Table 9.1,
- the identities (individual, male_parent, female_parent) are merged into a single list and the inverse relationship is formed before the data file is read,
- when the data file is read, data fields with the !P qualifier are recoded according to the combined identity list,
- the inverse relationship matrix is automatically associated with factors coded from the pedigree file unless some other covariance structure is specified. The inverse relationship matrix is specified with the variance model name AINV,
- the inverse relationship matrix is written to `ainverse.bin`,
 - if `ainverse.bin` already exists ASReml assumes it was formed in a previous run and has the correct inverse
 - `ainverse.bin` is read, rather than the inverse being reformed (unless !MAKE is specified); this saves time when performing repeated analyses based on a particular pedigree,
 - delete `ainverse.bin` or specify !MAKE if the pedigree is changed between runs,
- identities are printed in the `.sln` file,
 - identities should be whole numbers less than 200,000,000 unless ALPHA is specified,
 - pedigree lines for parents must precede their progeny,
 - unknown parents should be given the identity number 0,
 - if an individual appearing as a parent does not appear in the first column, it is assumed to have unknown parents, that is, parents with unknown parentage do not need their own line in the file,
 - identities may appear as both male and female parents, for example, in forestry.

9.5 Genetic groups

If all individuals belong to one genetic group, then use 0 as the identity of the parents of base individuals. However, if base individuals belong to various genetic groups this is indicated by the !GROUP qualifier and the pedigree file must begin by identifying these groups. All base individuals should have group identifiers as parents. In this case the identity 0 will only appear on the group identity lines, as in the following genetic group example.

```
Genetic group example
animal !P
sire 9 !A
dam
lines 2
damage
adailygain
harveyg.ped !ALPHA !MAKE !GROUP 3
harvey.dat
adailygain ~ mu
!r animal 02.5 !GU
```

```
G1 0 0
G2 0 0
G3 0 0
SIRE.1 G1 G1
SIRE.2 G1 G1
SIRE.3 G1 G1
SIRE.4 G2 G2
SIRE.5 G2 G2
SIRE.6 G3 G3
SIRE.7 G3 G3
SIRE.8 G3 G3
SIRE.9 G3 G3
101 SIRE.1 G1
102 SIRE.1 G1
103 SIRE.1 G1
:
163 SIRE.9 G3
164 SIRE.9 G3
165 SIRE.9 G3
```

Note that the sires in this example were from three lines. The same model can be fitted treating these as genetic groups, see Chapter 15.

Important

It is usually appropriate to allocate a genetic group identifier where the parent is unknown.

Table 9.1: List of pedigree file qualifiers

<i>qualifier</i>	description
!ALPHA	indicates that the identities are alphanumeric with up to 20 characters; otherwise by default they are numeric whole numbers < 200,000,000.
!GIV	instructs ASReml to write out the A-inverse in the format of .giv files.

List of pedigree file qualifiers

<i>qualifier</i>	description
!GROUPS <i>g</i>	includes genetic groups in the pedigree. The first <i>g</i> lines of the pedigree identify genetic groups (with zero in both the sire and dam fields). All other lines must specify one of the genetic groups as sire or dam if the actual parent is unknown.
!MAKE	tells ASReml to make the A-inverse (rather than trying to retrieve it from the <code>ainverse.bin</code> file).
!MGS	indicates that the third identity is the sire of the dam rather than the dam; a dam identity is created and interposed into the pedigree so that a pedigree line <code>animal sire mgs</code> is interpreted as <code>dam mgs 0</code> <code>animal sire dam</code> with a new dam being constructed for each such animal.
!REPEAT	tells ASReml to ignore repeat occurrences of lines in the pedigree file. Warning Use of this option will avoid the check that animals occur in chronological order, but chronological order is still required.
!SKIP <i>n</i>	allows you to skip <i>n</i> header lines at the top of the file.

9.6 Reading a user defined inverse relationship matrix

Sometimes an inverse relationship matrix is required other than the one ASReml can produce from the pedigree file. We call this a GIV (G inverse) matrix. The user can prepare a `.giv` file containing this matrix and use it in the analysis. The syntax for specifying a G inverse file named `name.giv` is

`name.giv` [`!SKIP n`]

- the named file must have a `.giv` extension,
- the G inverse files must be specified on the line(s) immediately prior to the data file line after any pedigree file,
- up to 6 G inverse matrices may be defined,
- the file must be free format with three numbers per line, namely

row column value

defining the lower triangle row-wise of the matrix,

- the file must be sorted *column* within *row*,
- every diagonal element must be represented; missing off-diagonal elements are assumed to be zero cells,
- the file is used by associating it with a factor in the model. The number and order of the rows must agree with the size and order of the associated factor,
- the `!SKIP n` qualifier tells ASReml to skip `n` header lines in the file.

```
1 1 1
2 2 1
3 3 1
4 4 1
5 5 1.0666667
6 5 -0.2666667
6 6 1.0666667
7 7 1.0666667
8 7 -0.2666667
8 8 1.0666667
9 9 1.0666667
10 9 -0.2666667
10 10 1.0666667
11 11 1.0666667
12 11 -0.2666667
12 12 1.0666667
```

The `.giv` file presented in the code box gives the following G inverse matrix

$$\begin{bmatrix} \mathbf{I}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_4 \otimes \begin{bmatrix} 1.067 & -0.267 \\ -0.267 & 1.067 \end{bmatrix} \end{bmatrix}$$

The `.giv` file can be associated with a factor in two ways:

- the first is to declare a G structure for the model term and to refer to the `.giv` file with the corresponding identifier GIV1, GIV2, GIV3, GIV4, GIV5, GIV6; for example,

```

animal 1
animal 0 GIV1 0.12

```

for a one-dimensional structure put the scale parameter (0.12 in this case) after the GIVg identifier,

```

site.variety 2
site 0 CORUH 0.5
8*1.5
variety 0 GIV1

```

for a two-dimensional structure.

- the second is for one-dimensional structures; in this case the `.giv` structure can be directly associated with the term using the `giv(f,i)` model function which associates the i th `.giv` file with factor f , for example,

```
giv(animal,1) 0.12
```

is equivalent to the first of the preceding examples.

The example continued

Below is an extension of `harvey.as` to include a `.giv` file. A portion of `harvey.giv` is presented to the right. In this case the G inverse matrix is simply an identity matrix of order 74 scaled by 0.5, that is, $0.5\mathbf{I}_{74}$. This model is simply an example which is easy to verify. Note that `harvey.giv` is specified on the line immediately preceding `harvey.dat`.

Model term specification associating the `harvey.giv` structure to the coding of sire takes precedence over the relationship matrix structure implied by the `!P` qualifier for sire. In this case, the `!P` is being used to amalgamate animals and sires into a single list.

command file

```

GIV file example
animal !P
sire !P
dam
lines 2
damage
adailygain
harvey.ped !ALPHA
harvey.giv # giv structure file
harvey.dat
adailygain ~ mu line, !r giv(sire,1) .25

```

.giv file

```

01 01 .5
02 02 .5
03 03 .5
04 04 .5
05 05 .5
.
.
.
72 72 .5
73 73 .5
74 74 .5

```


10

Tabulation of the data and prediction from the model

Introduction

Tabulation

Prediction

Examples

10.1 Introduction

Tabulation is the process of forming simple tables of averages and counts from the data. Such tables are useful for looking at the structure of the data and numbers of observations associated with factor combinations. Multiple `tabulate` directives may be specified in a job.

Prediction is the process of forming a linear function of the vector of fixed and random effects in the linear model to obtain an estimated or predicted value for a quantity of interest. It is primarily used for predicting tables of adjusted means. If a table is based on a subset of the explanatory variables then the other variables need to be accounted for. It is usual to form a predicted value either at specified values of the remaining variables, or averaging over them in some way.

This chapter describes the `tabulate` directive and the `predict` directive introduced in Section 3.4 under Prediction. However, we first describe the process and explain some of the concepts.

Our approach to prediction is a generalisation of that of Lane and Nelder (1982) who consider fixed effects models. They form fitted values for all combinations of the explanatory variables in the model, then take marginal means across the explanatory variables not relevant to the current prediction. Our case is more general in that random effects can be fitted in our (mixed) models. A full description can be found in Gilmour *et al.* (2002) and Welham *et al.* (2003).

Random factor terms may contribute to predictions in several ways. They may be evaluated at a values specified by the user, they may be averaged over, or they may be omitted from the fitted values used to form the prediction. Averaging over the set of random effects gives a prediction specific to the random effects observed. We describe this as a ‘conditional’ prediction. Omitting the term from the model produces a prediction at the population average (zero), that is, substituting the assumed population mean for an unknown random effect. We call this a ‘marginal’ prediction. Note that in any prediction, some terms may be evaluated as conditional and others at marginal values, depending on the aim of prediction.

For fixed factors there is no pre-defined population average, so there is no natural interpretation for a prediction derived by omitting a fixed term from the fitted values. Averages must therefore be taken over all the levels present to give a sample specific average, or value(s) must be specified by the user.

For covariate terms (fixed or random) the associated effect represents the coefficient of a linear trend in the data with respect to the covariate values. These terms should be evaluated at a given value of the covariate, or averaged over several given values. Omission of a covariate from the predictive model is equivalent to predicting at a zero covariate value, which is often inappropriate.

Interaction terms constructed from factors generate an effect for each combination of the factor levels, and behave like single factor terms in prediction. Interactions constructed from covariates fit a linear trend for the product of the covariate values and behave like a single covariate term. An interaction of a factor and a covariate fits a linear trend for the covariate for each level of the factor. For both fixed and random terms, a value for the covariate must be given, but the factor levels may be evaluated at a given level, averaged over or (for random terms only) omitted.

Before considering some examples in detail, it is useful to consider the conceptual steps involved in the prediction process. Given the explanatory variables used to define the linear (mixed) model, the four main steps are

(a) Choose the explanatory variable(s) and their respective value(s) for which predictions are required; the variables involved will be referred to as the *classify* set and together define the multiway table to be predicted.

(b) Determine which variables should be averaged over to form predictions. The values to be averaged over must also be defined for each variable; the variables involved will be referred to as the *averaging* set. The combination of the classify set with these averaging variables defines a multiway hyper-table. Note that variables evaluated at only one value, for example, a covariate at its mean value, can be formally introduced as part of the classifying or averaging set.

(c) Determine which terms from the linear mixed model are to be used in forming predictions for each cell in the multiway hyper-table in order to give appropriate conditional or marginal prediction.

(d) Choose the weights to be used when averaging cells in the hyper-table to produce the multiway table to be reported.

Note that after steps (a) and (b) there may be some explanatory variables in the fitted model that do not classify the hyper-table. These variables occur in terms that are ignored when forming the predicted values. It was deduced above that

fixed terms could not sensibly be ignored in forming predictions, so that variables should only be omitted from the hyper-table when they only appear in random terms. Whether terms derived from these variables should be used when forming predictions depends on the application and aim of prediction.

The main difference in this prediction process compared to that described by Lane and Nelder (1982) is the choice of whether to include or exclude model terms when forming predictions. In linear models, since all terms are fixed, terms not in the classify set must be in the averaging set.

10.2 Tabulation

A `tabulate` directive is provided to enable simple summaries of the data to be formed for the purpose of checking the structure of the data. Multiple `tabulate` statements are permitted and they must appear immediately after the linear model. The `tabulate` statement has the form

```
tabulate response_variables [!Wt weight !count !filter filter !select value]
~ factors
```

- `tabulate` is the directive name and must begin in column 1,
- `response_variables` is a list of variates for which means are required,
- `!Wt weight` nominates a variable containing weights,
- `!count` requests counts as well as means to be reported,
- `!filter filter` nominates a factor for selecting a portion of the data,
- `!select value` indicates that only records with `value` in the `filter` column are to be included,
- `~ factors` identifies the factors to be used for classifying the data. Only factors (not covariates) may be nominated and no more than six may be nominated.

ASReml prints the multiway table of means omitting empty cells to a file with extension `.tab`.

10.3 Prediction

The first step is to specify the classify set of explanatory variables after the `predict` directive. The `predict` statement(s) may appear immediately after the model line (before or after any `tabulate` statements) or after the R and G structure lines. The syntax is

```
predict factors [qualifiers]
```

- `predict` must be the first element of the `predict` statement, commencing in column 1 in upper or lower case,
- `factors` is a list of the variables defining a multiway table to be predicted; each variable may be followed by a list of specific values at which the prediction is to occur
- the `qualifiers` instruct ASReml to modify the predictions in some way, see Table 10.1 for a list of the prediction qualifiers,
- several `predict` statements may be specified.

```
NIN Alliance trial 1989
variety !A
:
column 11
nin89.asd !skip 1
yield ~ mu variety !r repl
predict variety
0 0 1
repl 1
repl 0 IDV 0.1
```

ASReml parses the `predict` statement before fitting the model. If any syntax problems are encountered, these are reported in the `.pvs` file after which the statement is ignored: the job is completed as if the erroneous prediction statement did not exist. The predictions are formed as an extra process in the final iteration and are reported to the `.pvs` file. Consequently, aborting a run by creating the `ABORTASR.NOW` file will cause any `predict` statements to be ignored.

By default, factors are predicted at each level, simple covariates are predicted at their overall mean and covariates used as a basis for splines or orthogonal polynomials are predicted at their design points. Model terms `mv` and `units` are always ignored.

Prediction at particular values of a covariate or particular levels of a factor is achieved by listing the values after the variate/factor name. Where there is a sequence of values, use the notation $a\ b \dots\ n$ to represent the sequence of values from a to n with step size $b - a$. The default stepsize is 1 (in which case b may be omitted). A colon (`:`) may replace the ellipsis (`...`). An increasing sequence is assumed. When giving particular values for factors, use the coded level ($1:n$) rather than the label (alphabetical or integer).

The second step is to specify the averaging set. The default averaging set is those explanatory variables involved in fixed effect model terms that are not in the classifying set. By default variables that only define random model terms are ignored. The qualifier **!AVERAGE** allows these variables to be added to the default averaging set.

The third step is to determine the linear model terms to use in prediction. The default rule is that all model terms based entirely on the classifying and averaging set are used. Two qualifiers allow this default set of model terms to be modified by adding (**!USE**) or removing (**!IGNORE**) terms. The qualifier **!ONLYUSE** explicitly specifies the model terms to use, ignoring all others. The qualifier **!EXCEPT** explicitly specifies the model terms not to use, including all others. These qualifiers may implicitly modify the averaging set by including variables defining terms in the predicted model not in the classify set. It is sometimes easier to specify the classify set and the prediction linear model and allow **ASReml** to construct the averaging set.

The fourth step is to choose the weighting for forming means over dimensions in the hyper-table. The default is to average over the specified levels but the qualifier **!AVERAGE factor weights** can be used to specify weights to be used in averaging over a factor.

There are often situations in which the fixed effects design matrix **X** is not of full column rank. These can be classified according to the cause of aliasing.

1. linear dependencies among the model terms due to over-parameterisation of the model,
2. no data present for some factor combinations so that the corresponding effects cannot be estimated,
3. linear dependencies due to other, usually unexpected, structure in the data.

The first type of aliasing is imposed by the parameterisation chosen and can be determined from the model. The second type of aliasing can be detected when setting up the design matrix for parameter estimation (which may require revision of imposed constraints). The third type can then be detected during the absorption of the mixed model equations. Dependencies (aliasing) can be dealt with in several ways and **ASReml** checks that predictions are of estimable functions in the sense defined by Searle (1971, p160) and are invariant to the

constraint method used.

Normally ASReml does not print predictions of non-estimable functions but the !PRINTALL qualifier instructs ASReml to print them.

If the data has missing combinations of factors there is the possibility that the corresponding cells of the hyper-table are not estimable and hence margins of the table are not estimable. In some cases it is useful to construct margins of tables based only on those cells that have data present. The qualifier !PRESENT does this.

The `predict` statement may be continued on subsequent lines by terminating the current line with a comma.

Table 10.1 is a list of the prediction qualifiers with the following syntax:

- f is an explanatory variable which is a factor,
- t is a list of terms in the fitted model,
- v is a list of explanatory variables.

Table 10.1: List of prediction qualifiers

<i>qualifier</i>	action
Controlling formation of tables	
!AVERAGE f [<i>weights</i>]	is used to formally include a variable in the averaging set and to explicitly set the weights for averaging. Variables that only appear in random model terms are not included in the averaging set unless included with this qualifier. The default for <i>weights</i> is equal weights. The list <i>weights</i> can have syntax like {3*1 0 2*1}/5 to represent the sequence 0.2 0.2 0.2 0 0.2 0.2. The string inside the curly brace is expanded first and the expression $n*v$ means n occurrences of v . A separate !AVERAGE qualifier is required for each variable requiring explicit weights or to be added to the default averaging set.
!PARALLEL [v]	without arguments means all classify variables are expanded in parallel. Otherwise up to four variables from the classify set may be specified.

List of prediction qualifiers

<i>qualifier</i>	action
<code>!PRESENT <i>v</i></code>	is used when averaging is to be based only on cells with data. <i>v</i> is a list of variables and may include variables in the classify set. <i>v</i> may not include variables with an explicit <code>!AVERAGE</code> qualifier. ASReml works out what combinations are present from the design matrix <i>W</i> . At present, the model must contain the factor as a simple main effect.
Controlling inclusion of model terms	
<code>!EXCEPT <i>t</i></code>	causes the prediction to include all fitted model terms not in <i>t</i> .
<code>!IGNORE <i>t</i></code>	causes ASReml to set up a prediction model based on the default rules and then removes the terms in <i>t</i> . This might be used to omit the spline Lack of fit term (<code>!IGNORE fac(x)</code>) from predictions as in <pre>yield ~ mu x variety !r spl(x) fac(x) predict x !IGNORE fac(x)</pre> which would predict points on the spline curve averaging over <i>variety</i> .
<code>!ONLYUSE <i>t</i></code>	causes the prediction to include only model terms in <i>t</i> . It can be used for example to form a table of slopes as in <pre>HI ~ mu X variety X.variety predict variety X 1 !onlyuse X X.variety</pre>
<code>!USE <i>t</i></code>	causes ASReml to set up a prediction model based on the default rules and then adds the terms listed in <i>t</i> .
Printing	
<code>!PLOT [<i>x</i>]</code>	instructs ASReml to attempt a plot of the predicted values. This qualifier is only applicable in versions of ASReml linked with the Interacter Graphics library. If there is no argument, ASReml uses the variable with the largest number of discrete values as the <i>x</i> variable in the graph. It draws a line for each combination of the other variables.
<code>!PRINTALL</code>	instructs ASReml to print the predicted value, even if it is not of an estimable function. By default, ASReml only prints predictions that are of estimable functions.
<code>!SED</code>	is specified to request all <i>standard errors of difference</i> be printed. Normally only an average value is printed.
<code>!VPV</code>	is specified to request that the variance matrix of predicted values be printed to the <code>.pvs</code> file.

Examples

Examples are as follows:

```
yield ~ mu variety !r repl  
predict variety
```

is used to predict variety means in the NIN field trial analysis. Random `repl` is ignored in the prediction.

```
yield ~ mu x variety !r repl  
predict variety
```

predicts variety means at the average of `x` ignoring random `repl`.

```
yield ~ mu x variety repl  
predict variety x 2
```

forms the hyper-table based on `variety` and `repl` at the covariate value of 2 and then averages across `repl` to produce variety predictions.

```
GFW Fdiam ~ Trait Trait.Year !r Trait.Team  
predict Trait Team
```

forms the hyper-table for each trait based on `Year` and `Team` with each linear combination in each cell of the hyper-table for each trait using `Team` and `Year` effects. `Team` predictions are produced by averaging over years.

```
yield ~ variety !r site.variety  
predict variety
```

will ignore the `site.variety` term in forming the predictions while

```
predict variety !AVERAGE site
```

forms the hyper-table based on `site` and `variety` with each linear combination in each cell using `variety` and `site.variety` effects and then forms averages across sites to produce variety predictions.

11

Functions of variance components

Introduction

Syntax

Linear combinations of components

Heritability

Correlation

A more detailed example

11.1 Introduction

ASReml includes a post-analysis procedure to calculate functions of variance components. Its intended use is when the variance components are either simple variances or are vari-

```
F phenvar 1 + 2 # pheno var
F genvar 1 * 4 # geno var
H herit 4 3 # heritability
```

ances and covariances in an unstructured matrix. The functions covered are linear combinations of the variance components (for example, phenotypic variance), a ratio of two components (for example, heritabilities) and the correlation based on three components (for example, genetic correlation). The user must prepare a `.pin` file. A simple sample `.pin` file is shown in the ASReml code box above. The `.pin` file specifies the functions to be calculated. The user re-runs ASReml with the `-P` command line option specifying the `.pin` file as the input file. ASReml reads the model information from the `.asr` and `.vvp` files and calculates the requested functions. These are reported in the `.pvc` file.

11.2 Syntax

Functions of the variance components are specified in the `.pin` file in lines of the form

letter label coefficients

- *letter* is either F, H or R as follows
 - F is for linear combinations of variance components,
 - H is for forming the ratio of two components,
 - R is for forming the correlation based on three components,
- *label* names the result,
- *coefficients* is the list of coefficients for the linear function.

Linear combinations of components

First ASReml extracts the variance components from the `.asr` file and their variance matrix from the `.vvp` file. Each linear function formed by an F line is added to the list of components. Thus, the number of coefficients increases by one each line. We seek to calculate $k + \mathbf{c}'\mathbf{v}$, $\text{cov}(\mathbf{c}'\mathbf{v}, \mathbf{v})$ and $\text{var}(\mathbf{c}'\mathbf{v})$ where \mathbf{v} is the vector of existing variance components, \mathbf{c} is the vector of coefficients for the linear function and k is an optional offset which is usually omitted but would be 1 to represent the

```
F phenvar 1 + 2 # pheno var
F genvar 1 * 4 # geno var
H herit 4 3 # heritability
```

residual variance in a probit analysis and 3.289 to represent the residual variance in a logit analysis. The general form of the directive is

```
F label a + b * c_b + c + d + m * k
```

where a , b , c and d are subscripts to existing components v_a , v_b , v_c and v_d and c_b is a multiplier for v_b . m is a number bigger than the current length of v to flag the special case of adding the offset k . Where matrices are to be combined the form

```
F label a:b * k + c:d
```

can be used, as in the Coopworth data example, see page 252.

Assuming that the `.pin` file in the ASReml code box corresponds to a simple sire model and that variance component 1 is the sire variance and variance component 2 is the residual variance, then

```
F phenvar 1 + 2
```

gives a third component which is the sum of the variance components, that is, the phenotypic variance, and

```
F genvar 1 * 4
```

gives a fourth component which is the sire variance component multiplied by 4, that is, the genotypic variance.

Heritability

Heritabilities are requested by lines in the `.pin` file beginning with an H. The specific form of the directive in this case is

```
F phenvar 1 + 2 # pheno var
F genvar 1 * 4 # geno var
H herit 4 3 # heritability
```

```
H label n d
```

This calculates σ_n^2/σ_d^2 and $se[\sigma_n^2/\sigma_d^2]$ where n and d are integers pointing to components v_n and v_d that are to be used as the numerator and denominator respectively in the heritability calculation.

$$\text{Var}\left(\frac{\sigma_n^2}{\sigma_d^2}\right) = \left(\frac{\sigma_n^2}{\sigma_d^2}\right)^2 \left(\frac{\text{Var}(\sigma_n^2)}{\sigma_n^4} + \frac{\text{Var}(\sigma_d^2)}{\sigma_d^4} - \frac{2\text{Cov}(\sigma_n^2, \sigma_d^2)}{\sigma_n^2 \sigma_d^2} \right)$$

In the example

```
H herit 4 3
```

calculates the heritability by calculating component 4 (from second line of `.pin`)

/ component 3 (from first line of `.pin`), that is, genetic variance / phenotypic variance.

Correlation

Correlations are requested by lines in the `.pin` file beginning with an `R`. The specific form of the directive is

```
F phenvar 1:3 + 4:6
R phencorr 7 8 9
R gencorr 4:6
```

`R label a ab b`

This calculates the correlation $r = \sigma_{ab} / \sqrt{\sigma_a^2 \sigma_b^2}$ and the associated standard error. a , b and ab are integers indicating the position of the components to be used. Alternatively,

`R label aa:nn`

calculates the correlation $r = \sigma_{ab} / \sqrt{\sigma_a^2 \sigma_b^2}$ for all correlations in the lower triangular row-wise matrix represented by components aa to nn and the associated standard errors.

$$\begin{aligned} \text{var}(r) = r^2 & \left[\frac{\text{var}(\sigma_a^2)}{4\sigma_a^2} + \frac{\text{var}(\sigma_b^2)}{4\sigma_b^2} + \frac{\text{var}(\sigma_{ab})}{\sigma_{ab}^2} \right. \\ & \left. + \frac{2\text{cov}(\sigma_a^2)\sigma_b^2}{4\sigma_a^2\sigma_b^2} - \frac{2\text{cov}(\sigma_a^2)\sigma_{ab}}{2\sigma_a^2\sigma_{ab}} - \frac{2\text{cov}(\sigma_{ab})\sigma_b^2}{2\sigma_{ab}\sigma_b^2} \right] \end{aligned}$$

In the example

```
R phencorr 7 8 9
```

calculates the phenotypic covariance by calculating component 8 / $\sqrt{\text{component 7} \times \text{component 9}}$ where components 7, 8 and 9 are created with the first line of the `.pin` file, and

```
R gencorr 4:6
```

calculates the genotypic covariance by calculating component 5 / $\sqrt{\text{component 4} \times \text{component 6}}$ where components 4, 5 and 6 are variance components from the analysis.

A more detailed example

The example `.pin` file in this code box is a little more complicated. It relates to the **bivariate sire model** in `bsiremod.as` shown below.

```
F phenvar 1:3 + 4:6
F addvar 4:6 * 4
H heritA 10 7
H heritB 12 9
R phencorr 7 8 9
R gencorr 4:6
```

```

:
:
.asr file
```

Source	Model terms	Gamma	Component	Comp/SE	% C
Residual	UnStruct	1 26.2197	26.2197	18.01	0 U
Residual	UnStruct	1 2.85090	2.85090	9.55	0 U
Residual	UnStruct	2 1.71556	1.71556	18.00	0 U
Tr.sire	UnStruct	1 16.5262	16.5262	2.69	0 U
Tr.sire	UnStruct	1 1.14422	1.14422	1.94	0 U
Tr.sire	UnStruct	2 0.132847	0.132847	1.88	0 U

```

:
:
```

The ASReml file `bsiremod.as` is presented in the code box to the right. Above is the section of the `.asr` file that displays the variance component table on which the `.pin` file is based.

```
Bivariate sire model
sire !I
ywt fat
bsiremod.asd
ywt fat ~ Trait !r Trait.sire
1 2 1
0 # ASReml will count units
Trait 0 US
3*0
Trait.sire 2
Trait 0 US
3*0
sire
```

Numbering the parameters reported in `bsiremod.asr` (and `bsiremod.vvp`)

- 1 error variance for `ywt`
- 2 error covariance for `ywt` and `fat`
- 3 error variance for `fat`
- 4 sire variance component for `ywt`
- 5 sire covariance for `ywt` and `fat`
- 6 sire covariance for `fat`

then

```
F phenvar 1:3 + 4:6
```

creates new components $7 = 1+4$, $8 = 2+5$
and $9 = 3+6$,

```
F addvar 4:6 * 4
```

creates new components $10 = 4 \times 4$, $11 = 5 \times 4$ and $12 = 3 \times 4$,

```
H heritA 10 7
```

forms $10 / 7$ to give the heritability for `ywt`,

```
H heritB 12 9
```

forms $12 / 9$ to give the heritability for `fat`,

```
R phencorr 7 8 9
```

forms $8 / \sqrt{7 \times 9}$, that is, the phenotypic correlation between `ywt` and `fat`,

```
R gencorr 4:6
```

forms $5 / \sqrt{4:6}$, that is, the genetic correlation between `ywt` and `fat`.

The result is:

`.pvc file`

```

7 phenvar 1      42.75      6.297
8 phenvar 2      3.995      0.6761
9 phenvar 3      1.848      0.1183
10 addvar 4      66.10      24.58
11 addvar 5      4.577      2.354
12 addvar 6      0.5314     0.2831
h2ywt      = addvar 10/phenvar 7=      1.5465  0.3574
h2fat      = addvar 12/phenvar 9=      0.2875  0.1430
phencorr   = phenvar /SQR[phenvar *phenvar ]= 0.4495  0.0483
gencor 2 1 = Tr.si 5/SQR[Tr.si 4*Tr.si 6]= 0.7722  0.1537
```

12 Command file: Running the job

Introduction

The command line

Command line options

- Debug command line options
- Graphics command line options
- Job control command line options
- Memory command line options
- Menu command line options
- Non-graphics command line options
- Post-processing calculation of variance components
- Examples

Advanced processing arguments

- Standard use of arguments
- Entering command line arguments interactively

12.1 Introduction

The command line is discussed in this chapter. Command line options enable more memory to be accessed to run the job, control some graphics output, enable use of a simple menu and control advanced processing options.

12.2 The command line

The basic command to run ASReml is

```
[path] ASReml basename [.as[c]]
```

- *path* provides the path to the ASReml program which is usually called `asrem1.exe` in a PC environment. In a UNIX environment, ASReml is usually run through a shell script called `ASReml`.

– if the ASReml program is in the search path then *path* is not required and the word `ASReml` will suffice; for example

```
ASReml nin89.as
```

will run the NIN analysis,

– if `asrem1.exe` (ASReml) is not in the search path then *path* is required, for example, if `asrem1.exe` is in the subdirectory `c:\Asrem1` then

```
c:\Asrem1\asrem1 nin89.as
```

will run `nin89.as`,

- ASReml invokes the ASReml program,
- *basename* is the name of the `.as[c]` command file.

The basic command line can be extended with options and arguments to

```
[path] ASReml [options] basename [.as[c]] [arguments]
```

- *options* is a string preceded by a - (minus) sign. Its components control several operations (batch, graphic, memory, ...) at run time; for example, the command line

```
ASReml -s4 rat.as
```

tells ASReml to run the job `rat.as` with memory allocation increased to 128mb,

- *arguments* provide a mechanism (mostly for advanced users) to modify a job at run time; for example, the command line

```
ASReml rat.as alpha beta
```

tells ASReml to process the job in `rat.as` as if it read `alpha` wherever `$1` appears in the file `rat.as`, `beta` wherever `$2` appears and `0` wherever `$3` appears (see below).

12.3 Command line options

Table 12.1 presents the command line options available in ASReml with brief descriptions. Detailed descriptions follow. Command line options are not case sensitive.

Table 12.1: Command line options

<i>option</i>	type	action
Frequently used command line options		
C	job control	continue iterations using previous estimates as initial values
F	job control	continue for one more iteration using previous estimates as initial values
L	non-graphics output	direct screen output to file <i>basename.asl</i>
N	graphics	suppress interactive graphics
Ss	memory	control available memory
Other command line options		
D	debug	invoke debug mode
E	debug	invoke extended debug mode
Gg	graphics	set interactive graphics device
I	graphics	permit interactive graphics to be displayed
M	menu	request menu mode
P	post-processing	calculation of function of variance components
R	job control	repeat run for each argument

Command line options should be combined in a single string preceded by a -

(minus) sign. For example

`-LNS3`

could be used for this combination of options.

Debug command line options (D, E)

D and E invoke debug mode and increase the information written to the screen or `.asl` file. This information is not useful to most users. On **Unix** systems, if **ASReml** is crashing use the system `script` command to capture the screen output rather than using the L option, as the `.asl` file is not properly closed after a crash.

Graphics command line options (G, I, N)

The I option permits the variogram and residual graphics to be displayed. This is the default unless the L option is specified.

The N option prevents any graphics from being displayed. This is also the default when the L option is specified.

The `Gg` option sets the file type for hard copy versions of the variogram and residual graphics. Hard copy is only formed if the graphics are displayed.

Graphics are produced in the PC and SUN versions of **ASReml** using the Interacter graphics library. **ASReml** writes the graphics to file `aspi.e` where *p* indicates the part of the job, *i* indicates the contents of the figure (as given in the following table) and *e* indicates the file type, for example `ps` for postscript:

<i>i</i>	file contents
<code>._1_R</code>	marginal means of residuals from spatial analysis for section 1
<code>._2_V</code>	variogram of residuals from spatial analysis for section 2
<code>._1_S</code>	residuals in field plan
<code>._A_H</code>	histogram of residuals
<code>Rv_E</code>	residuals plotted against expected values
<code>XY_G</code>	figure produced by !X, !Y and !G qualifiers
<code>._PVa</code>	Predicted values plotted

The file type can be specified by following the G option by a number *g*, as follows:

<i>g</i>	description
1	HP-GL
2	Postscript (default for Unix)
6	BMP
10	Windows Print Manager
11	WinMetaFile (default for Windows)
12	HP-GL2
22	EncapsulatedPostScript

Not all of these file types are available on all systems but 2, 12 and 22 should all be available. The file type may also be set from the user menu by selecting **Opt Form** from the menu.

Job control command line options (C, F, R)

C indicates that the job is to continue iterating from the values in the `.rsv` file. This is equivalent to setting the `!CONTINUE` data file line qualifier, see Table 5.4 for details.

F indicates that the job is to continue for one more iteration from the values in the `.rsv` file. This is useful when using `predict`, see Chapter 10.

R runs `ASReml` once for each argument on the command line (see page 148) and includes the argument in the output file name, see the multivariate animal genetics example of Section 15.10 at page 250.

Memory command line options (S)

`Ss` is used to increase memory allocation to the program. The base memory allocated is approximately 8mb (1 million double precision words). `s` is a digit. The memory allocated is 2^s times the base amount. The sequence is 8, 32, 64, 128, 256, 512, 1024,...

If your system cannot provide the requested memory, the request will be diminished until it can be satisfied. On multi-user systems, do not make `s` larger than necessary or other users may complain.

The memory requirements depend on problem size and may be quite large. If `Ss` is not active (it is compiler dependent) you will need to request a version of `ASReml` compiled with a larger workspace.

Menu command line options (M)

M runs ASReml in menu mode. This mode is only available for the Microsoft Windows, Intel Linux and Sun Solaris versions of ASReml as these are linked with the Interacter graphics library. Menu mode is particularly intended to make ASReml easier to use under Windows. The menu enables you to edit the command file, run the job and view the output from within ASReml. Otherwise these are separate operations from the command line. Menu mode is suitable for regular analyses but large/long runs are better run from the command line as information reporting the progress of the job is not readily visible when running in menu mode. The primary menu options are

- File
 - when menu mode is invoked without a filename argument, the File option is invoked to input a file name,
 - normally you will open an existing .as command file but you may nominate a .dat, .asd or .csv file in which case ASReml will create a .as template file for you to edit
 - this is helpful if the data file contains field names on the first line,
 - the .as file is displayed in an edit window,
- Edit
 - invokes a primitive editor for you to edit the .as file and then run the job; you must explicitly save the file before running it,
- Run
 - this runs the current job and returns the .asl (if L option is active) or .asr file to a scroll screen for the user to scan or print,
- View
 - this invokes a submenu which allows you to browse the .asr, .asl, .sln, .res or .pvs files,
 - the browse menu allows you to search for particular text strings,
- Opt
 - displays the current state of the C, D, E, G, L and R command line options and allows you to toggle their state,
 - the graphics line shows the current hard copy graphics file type,
- Help
 - displays a syntax summary in a browse window,

- Exit
 - closes ASReml down,
- Quit
 - returns to the previous menu,
- Info
 - describes the main menu.

Log file (L)

L causes most of the screen output to be directed to a file with the file extension `.asl`. This is desirable for batch processing. It also suppresses the interactive graphics (available for spatial analyses) unless the I option is also specified.

Post-processing calculation of variance components (P)

P invokes the post-processing sub-program, see Chapter 11; this sub-program is used to calculate functions of variance components from the information in the `.asr` and `.vvp` files and it is run after the model has been fitted.

Examples

ASReml code	action
<code>asreml -L rat.as</code>	send screen output to <code>rat.as1</code> and suppress interactive graphics
<code>asreml -LS3 rat.as</code>	increase memory to 8 ($= 2^3$) times the base amount
<code>asreml -IL rat.as</code>	send screen output to <code>rat.as1</code> but display interactive graphics
<code>asreml -N rat.as</code>	allow screen output but suppress interactive graphics
<code>asreml -ILS2 rat.as</code>	increase memory to 4 times the base amount, send screen output to <code>rat.as1</code> but display interactive graphics
<code>asreml -MS2 rat.as</code>	invoke Menu model to edit and run the <code>rat.as</code> job using 32Mb workspace

12.4 Advanced processing arguments

Standard use of arguments

Command line arguments are intended to facilitate the running of a sequence of jobs that require small changes to the command file between runs. The output file name is modified by the use of this feature if the `-R` option is specified. This use is demonstrated in the Coopworth example of Section 15.10, see page 250.

Command line arguments are strings listed on the command line after *basename*, the command file name. These strings are inserted into the command file at run time. When the input routine finds a $\$n$ in the command file it substitutes the n th argument string from the command line. n may take the values 1..9 to indicate up to 9 strings after the command file name. If the argument has 1 character a trailing blank is attached to the character and inserted into the command file. If no argument exists, a zero is inserted. For example,

```
asreml rat.as alpha beta
```

tells ASReml to process the job in `rat.as` as if it read `alpha` wherever $\$1$ appears in the command file, `beta` wherever $\$2$ appears and 0 wherever $\$3$ appears.

Table 12.2: The use of arguments in ASReml

in command file	on command line	becomes in ASReml run
abc\$1def	no argument	abc0 def
abc\$1def	with argument X	abcX def
abc\$1def	with argument XY	abcXYdef
abc\$1def	with argument XYZ	abcXYZdef
abc\$1 def	with argument XX	abcXX def
abc\$1 def	with argument XXX	abcXXX def
abc\$1 def (multiple spaces)	with argument XXX	abcXXX def

Recycle

Recycle mode is invoked with the R option. Assuming there are say 4 arguments 11 22 33 44, the job is run 4 times: in the first run the arguments are 11 22 33 44; in the second run the arguments are 22 33 44; in the third run the arguments are 33 44; in the fourth run the argument is 44. Usually this mode is used with \$1 in the command file so that only the first of the arguments is honoured in a particular run. This is mainly used in conjunction with the !DOPART qualifier. Note that the names of the output files will be formed using a root based on *basename* and the current first argument so that they are not overwritten.

Entering command line arguments interactively

Another way to gain some interactive control of a job in the PC environment is to insert !?*text* in the .as file where you want to specify the rest of the line at run time. ASReml prompts with *text* and waits for a response which is used to complete the line. The !? qualifier may be used anywhere in the job and the line is modified from that point.

Warning

Unfortunately the prompt may not appear on the top screen under some windows operating systems in which case it may not be obvious that ASReml is waiting for a keyboard response.

13

Description of output files

Introduction

An example

Key output files

The .asr file
The .sln file
The .yht file

Other ASReml output files

The .res file
The .vrb file
The .vvp file
The .rsv file
The .dpr file
The .pvc file
The .pvs file
The .tab file

ASReml output objects and where to find them

13.1 Introduction

With each ASReml run a number of output files are produced. ASReml generates the output files by appending various filename extensions to *basename*. A brief description of the filename extensions is presented in Table 13.1. The `.sln` and `.yht` files, in particular, are often quite large and could fill up your disk space. You should therefore regularly tidy your working directories.

Table 13.1: Summary of ASReml output files

file	description
Key output files	
<code>.asr</code>	contains a summary of the data and analysis results.
<code>.sln</code>	contains the estimates of the fixed and random effects and their corresponding standard errors.
<code>.yht</code>	contains the predicted values, residuals and diagonal elements of the hat matrix for each data point.
Other output files	
<code>.asl</code>	contains a progress log and error messages if the <code>L</code> command line options is specified.
<code>.asp</code>	contains transformed data, see <code>!PRINT</code> in Table 5.2.
<code>.dbr/.dpr/.spr</code>	contains the data and residuals in a binary form for further analysis (see <code>!RESIDUALS</code> , Table 5.5).
<code>.pvc</code>	contains the report produced with the <code>P</code> option.
<code>.pvs</code>	contains predictions formed by the <code>predict</code> directive.
<code>.res</code>	contains information from using the <code>pol()</code> , <code>spl()</code> and <code>fac()</code> functions, the iteration sequence for the variance components and some statistics derived from the residuals to assist with model selection.
<code>.rsv</code>	contains the final parameter values for reading back if the <code>!CONTINUE</code> qualifier is invoked, see Table 5.4.
<code>.tab</code>	contains tables formed by the <code>tabulate</code> directive.
<code>.veo</code>	holds the equation order to speed up re-running big jobs when the model is unchanged. This is a binary file of no use to the user.
<code>.vrb</code>	contains the estimates of the fixed effects and their variance.
<code>.vvp</code>	contains the approximate variances of the variance parameters. It is designed to be read back with the <code>P</code> option for calculating functions of the variance parameters.

Summary of ASReml output files

file	description
------	-------------

13.2 An example

In this chapter the ASReml output files are discussed with reference to a two-dimensional separable autoregressive spatial analysis of the NIN field trial data, see model **3b** on page 83 of Chapter 7 for details. The ASReml command file for this analysis is presented to the right. Recall that this model specifies a separable autoregressive correlation structure for residual or plot errors that is the direct product of an autoregressive correlation matrix of order 22 for rows and an autoregressive correlation matrix of order 11 for columns. In this case 0.3 is the starting correlation for both columns and rows.

```
NIN Alliance Trial 1989
variety !A
id
raw
repl 4
nloc
yield
lat
long
row 22
column 11
nin89.asd !skip 1
yield ~ mu variety !f mv
predict variety
1 2 0
22 row AR1 0.3
11 column AR1 0.3
```

13.3 Key output files

The key ASReml output files are the `.asr`, `.sln` and `.yht` files.

The `.asr` file

This file contains

- a general announcements box (outlined in asterisks) containing current messages,
- a summary of the data to validate the specification of the model,
- a summary of the fitting process to check convergence,
- a summary of the variance parameters:
 - The Gamma column reports the actual parameter fitted,

- the Component column reports the gamma converted to a variance scale if appropriate,
 - Comp/SE is the ratio of the component relative to the square root of the diagonal element of the inverse of the average information matrix [Warning](#) Comp/SE should not be used for formal testing,
 - The % shows the percentage change in the parameter at the last iteration,
 - use the .pin file described Chapter 11 to calculate meaningful functions of the variance components,
- an analysis of variance (ANOVA) table (Section 6.10). The table contains the numerator degrees of freedom for the term and F-tests for approximate testing of effects. This is incremental test adjusted for *preceding terms only*.
 - estimated effects, their standard errors and *t* values for equations in the DENSE portion of the SSP matrix; the T-prev column tests difference between successive coefficients in the same factor.

The following is a truncated version of `nin89.asr`.

```

version & title   ASReml 1.0 [31 Aug 2002]  nin alliance trial
date & time      10 Sep 2002 04:20:15.350  8.00 Mbyte  Windows  nin89
license         Licensed to: Arthur Gilmour
announcements   * ASReml 2002 *****
box            * Contact asreml@vsn-intl.com for licensing and support *
               * Use !{ and !} when fitting covariance across terms *
               ***** ARG *
               Folder: C:\asr\ex\manex
               QUALIFIERS: !SKIP 1
               Reading nin.asd FREE FORMAT skipping      1 lines
data summary    Univariate analysis of yield
               Using      242 records of      242 read
               Model term      Size Type  COL  Minimum  Mean      Maximum  #zero #miss
               1 variety        56 Factor  1      1      26.4545   56        0      0
               2 id             2 1.000  2 1.000  26.45     56.00     0      0
               3 pid           3 1101.  3 1101.  2628.     4156.     0     18
               4 raw           4 21.00  4 21.00  510.5     840.0     0     18
               5 repl          4 Factor  5      1      2.4132    4         0      0
               6 nloc          6 4.000  6 4.000  4.000     4.000     0      0
               7 yield         1 Variate 7 1.050  25.53     42.00     0     18
               8 lat           8 4.300  8 4.300  25.80     47.30     0      0

```

```

      9 long                9  1.200      13.80      26.40      0  0
     10 row                22 Factor  10    1    11.5000      22      0  0
     11 column            11 Factor  11    1     6.0000      11      0  0
     12 mu                  1 Constant Term
     13 mv_estimates      18 Missing value
        22 AR=AutoReg      0.3000
        11 AR=AutoReg      0.3000
Forming      75 equations:  57 dense.
Initial updates will be shrunk by factor  0.316
NOTICE:      1 (more) singularities,
iterations
sequence      1 LogL=-419.949      S2=  39.387      168 df      1.000      0.3000      0.3000
              2 LogL=-410.420      S2=  38.350      168 df      1.000      0.4030      0.3461
              3 LogL=-402.549      S2=  40.253      168 df      1.000      0.5202      0.3999
              4 LogL=-399.465      S2=  46.228      168 df      1.000      0.6233      0.4423
              5 LogL=-399.331      S2=  48.139      168 df      1.000      0.6475      0.4414
              6 LogL=-399.324      S2=  48.580      168 df      1.000      0.6535      0.4389
              7 LogL=-399.324      S2=  48.680      168 df      1.000      0.6550      0.4379
              8 LogL=-399.324      S2=  48.705      168 df      1.000      0.6554      0.4376
Final parameter values                1.0000      0.65549      0.43750

variance
parameters
Source          Model terms      Gamma      Component      Comp/SE      % C
Variance                242  168  1.00000      48.7045      6.81  0 P
Residual            AR=AutoR      22  0.655488      0.655488      11.63  0 U
Residual            AR=AutoR      11  0.437497      0.437497      5.43  0 U

ANOVA
Analysis of Variance                DF      F-incr
  12 mu                1      331.90
   1 variety            55      2.23
NOTICE: ASReml cannot yet determine the correct denominator DFs for
        these F statistics.  They will not exceed the Residual DF
        which applies only if all components are KNOWN.

              Estimate      Standard Error      T-value      T-prev
estimated
effects      1 variety
BRULE                2.98387      2.84137      1.05
REDLAND              4.70621      2.97738      1.58      0.62
CODY                 -0.316046      2.96066      -0.11      -1.76
ARAPAHOE             2.95377      2.72658      1.08      1.16
NE83404              1.63070      2.89322      0.56      -0.48
NE83406              1.29029      2.77143      0.47      -0.12
NE83407              0.308992      3.13838      0.10      -0.31
CENTURA             2.26393      3.00167      0.75      0.66
:
NE87615              1.03251      2.93379      0.35      -0.40
NE87619              5.93687      2.84941      2.08      1.82
NE87627              -4.37848      2.99740      -1.46      -3.72
  12 mu
                    57  24.0892      2.46480      9.77
  13 mv_estimates                18 effects fitted
outlier info          6 possible outliers: see .res file
completion info Finished: 10 Sep 2002 04:20:34.508  LogL Converged

```

The .sln file

The .sln file contains estimates of the fixed and random effects with their standard errors in an array with four columns ordered as

```
factor_name level estimate standard_error
```

Note that the error presented for the estimate of a random effect is the square root of the prediction error variance. The .sln file can easily be read into a GENSTAT spreadsheet or an S-PLUS data frame. This is a reduced version of nin89.sln. Note that

- the order of some terms may differ from the order in which those terms were specified in the model statement,
- the missing value estimates appear at the end of the file in this example.

```
variety estimates  variety          LANCER          0.000          0.000
                   variety          BRULE            2.987          2.842
                   variety          REDLAND         4.707          2.978
                   variety          CODY           -0.3131         2.961
                   variety          ARAPAHOE        2.954          2.727
                   :
                   variety          NE87615         1.035          2.934
                   variety          NE87619         5.939          2.850
                   variety          NE87627        -4.376          2.998
intercept          mu                    1            24.09          2.465
missing value     mv_estimates          1            21.91          6.729
estimates         mv_estimates          2            23.22          6.721
                   mv_estimates          3            22.52          6.708
                   mv_estimates          4            23.49          6.676
                   mv_estimates          5            22.26          6.698
                   mv_estimates          6            24.47          6.707
                   mv_estimates          7            20.14          6.697
                   mv_estimates          8            25.01          6.691
                   mv_estimates          9            24.29          6.676
                   mv_estimates         10            26.30          6.658
                   mv_estimates         11            24.99          6.590
                   mv_estimates         12            27.78          6.492
                   :
```

The .yht file

The .yht file contains the predicted values of the data in the original order (this is not changed by supplying row/column order in spatial analyses), the residuals and the diagonal elements of the hat matrix. Where an observation is missing, the residual, missing values predicted value and **Hat** value are also declared missing. The missing value estimates with standard errors are reported in the .sln file.

This is the first 20 lines of nin89.yht. Note that the values corresponding to the missing data (first 15 records) are all $-0.1000E-36$ which is the internal value used for missing values.

Record	Yhat	Residual	Hat
1	-0.1000E-36	-0.1000E-36	-0.1000E-36
2	-0.1000E-36	-0.1000E-36	-0.1000E-36
3	-0.1000E-36	-0.1000E-36	-0.1000E-36
4	-0.1000E-36	-0.1000E-36	-0.1000E-36
5	-0.1000E-36	-0.1000E-36	-0.1000E-36
6	-0.1000E-36	-0.1000E-36	-0.1000E-36
7	-0.1000E-36	-0.1000E-36	-0.1000E-36
8	-0.1000E-36	-0.1000E-36	-0.1000E-36
9	-0.1000E-36	-0.1000E-36	-0.1000E-36
10	-0.1000E-36	-0.1000E-36	-0.1000E-36
11	-0.1000E-36	-0.1000E-36	-0.1000E-36
12	-0.1000E-36	-0.1000E-36	-0.1000E-36
13	-0.1000E-36	-0.1000E-36	-0.1000E-36
14	-0.1000E-36	-0.1000E-36	-0.1000E-36
15	-0.1000E-36	-0.1000E-36	-0.1000E-36
16	24.088	5.162	6.074
17	27.074	4.476	6.222
18	28.795	6.255	6.282
19	23.775	6.325	6.235
20	27.042	6.008	5.962
⋮			
240	24.695	1.855	6.114
241	25.452	0.1475	6.158
242	22.465	4.435	6.604

13.4 Other ASReml output files

The .res file

The .res file contains miscellaneous supplementary information including

- a list of unique values of x formed by using the `fac()` model term,
- a list of unique (x, y) combinations formed by using the `fac(x, y)` model term,
- orthogonal polynomials produced by `pol()` model term,
- the design matrix formed for the `spl()` model term,
- predicted values of the curvature component of cubic smoothing splines,
- the empirical variance-covariance matrix based on the BLUPs when a $\Sigma \otimes \mathbf{I}$ or $\mathbf{I} \otimes \Sigma$ structure is used; this may be used to obtain starting values for another run of ASReml,
- a table showing the variance components for each iteration,
- some statistics derived from the residuals from two-dimensional data (multivariate, repeated measures or spatial)
 - the residuals from a spatial analysis will have the `units` part added to them (defined as the combined residual) unless the data records were sorted (within ASReml) in which case the `units` and the correlated residuals are in different orders (data file order and field order respectively),
 - the residuals are printed in the `.yht` file but the statistics in the `.res` file are calculated from the combined residual,
 - the `Covariance/Variance/Correlation (C/V/C)` matrix calculated directly from the residuals; it contains the covariance below the diagonals, the variances on the diagonal and the correlations above the diagonal - these may help with finding starting values,
 - relevant portions of the estimated variance matrix for each term for which an R structure or a G structure has been associated,
- a variogram and spatial correlations for spatial analysis; the spatial correlations are based on distance between data points (see Gilmour *et al.*, 1997),
- the slope of the $\log(\text{absolute residual})$ on $\log(\text{predicted value})$ for assessing possible mean-variance relationships and the location of large residuals. For example,

SLOPES FOR LOG(ABS(RES)) ON LOG(PV) for section 1

0.99 2.01 4.34

produced from a trivariate analysis reports the slopes. A slope of b suggests

that y^{1-b} might have less mean variance relationship. If there is no mean variance relation, a slope of zero is expected. A slope of $\frac{1}{2}$ suggests a SQRT transformation might resolve the dependence; a slope of 1 means a LOG transformation might be appropriate. So, for the 3 traits, $\log(y_1)$, y_2^{-1} and y_3^{-3} are indicated. This diagnostic strategy works better when based on grouped data regressing $\log(\text{standard deviation})$ on $\log(\text{mean})$.

Also,

```
STND RES 16 -2.35 6.58 5.64
```

indicates that for the 16th data record, the residuals are -2.35, 6.58 and 5.64 times the respective standard deviations. The standard deviation used in this test is calculated directly from the residuals rather than from the analysis. They are intended to flag the records with large residuals rather than to precisely quantify their relative size. They are not studentised residuals and are generally not relevant when the user has fitted heterogeneous variances.

This is nin89.res.

C/V/C matrix

Covariance/Variance/Correlation matrix of Residuals								Section 1
18.261	0.522	0.688	0.684	0.819	0.496	0.445	0.499	
0.053	0.083	0.180						
14.887	44.523	0.710	0.550	0.517	0.215	0.568	0.550	
0.439	0.263	0.318						
14.934	24.080	25.811	0.626	0.727	0.494	0.534	0.587	
0.332	0.185	0.329						
14.193	17.818	15.433	23.546	0.735	0.479	0.587	0.303	
0.055	-0.059	0.059						
21.332	21.041	22.519	21.751	37.185	0.345	0.501	0.540	
0.212	0.236	0.279						
14.092	9.539	16.664	15.431	13.984	44.158	0.614	0.033	
-0.314	-0.482	-0.417						
13.350	26.637	19.067	20.022	21.461	28.647	49.371	0.435	
0.173	-0.037	-0.053						
22.425	38.622	31.364	15.478	34.628	2.300	32.134	110.587	
0.762	0.721	0.778						
3.040	39.560	22.759	3.613	17.429	-28.133	16.449	108.223	
182.312	0.887	0.885						
4.373	21.628	11.579	-3.524	17.764	-39.584	-3.237	93.638	
147.844	152.427	0.921						
10.074	27.827	21.927	3.767	22.284	-36.277	-4.919	107.276	
156.540	148.997	171.732						

Fitted Covariance/Variance matrix of Residuals					is AR=AutoReg		
1.000	0.438	0.192	0.084	0.037	0.016	0.007	0.003
0.001	0.001	0.000					

The sample variogram is a plot of the semi-variances of differences of residuals at particular distances. The (0,0) position is zero because the difference is identically zero. ASReml displays the plot for distances 0, 1, 2, ..., 8, 9-10, 11-14, 15-20,

The plot of residuals in field plan order (Figure 13.2) contains in its top and right margins a diamond showing the minimum, mean and maximum residual for that row or column. Note that a gap identifies where the missing values occur.

The plot of marginal means of residuals shows residuals for each row/column as well as the trend in their means.

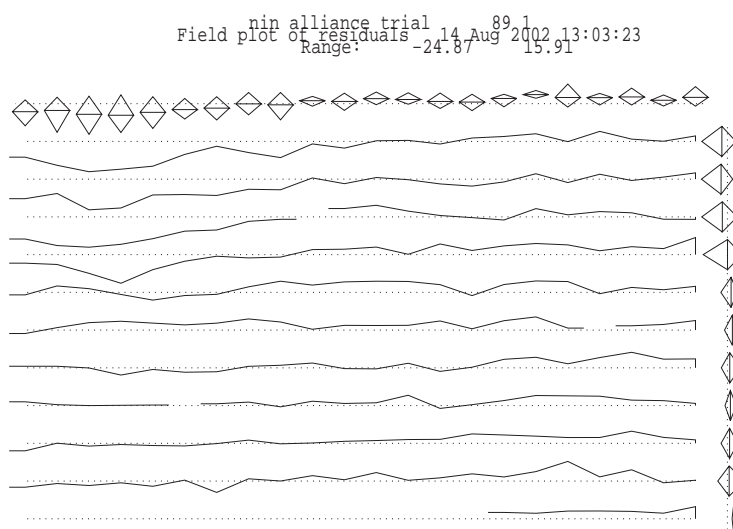


Figure 13.2 Plot of residuals in field plan order

The .vrb file

The .vrb file contains the estimates of the effects together with their approximate prediction variance matrix corresponding to the dense portion. The file is formatted for reading back for post processing. The number of equations in the dense portion can be increased (to a maximum of 800) using the !DENSE option (Table 5.5) but not to include random effects. The matrix is lower triangular row-wise in the order that the parameters are printed in the .sln file. It can be thought of as a partitioned lower triangular matrix,

$$\begin{bmatrix} \sigma^2 & . \\ \tilde{\beta}_D & \sigma^2 C^{DD} \end{bmatrix}$$

nin alliance trial 89 1
 Residuals V Row and Column position: 14 Aug 2002 13:03:23
 Range: -24.87 15.91

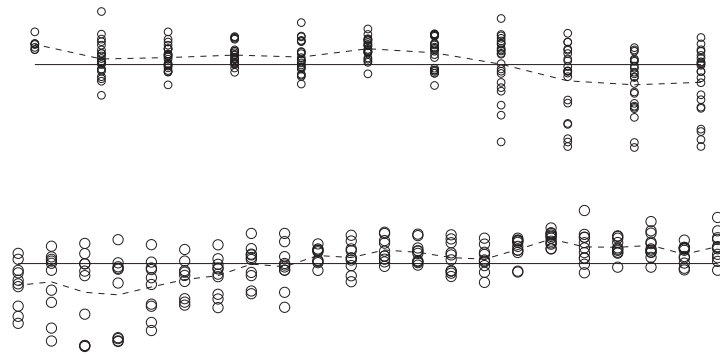


Figure 13.3 Plot of the marginal means of the residuals

nin alliance trial 1 A
 Histogram of residuals: 14 Aug 2002 13:03:23

Peak Count: 17
 Range: -24.87 15.91

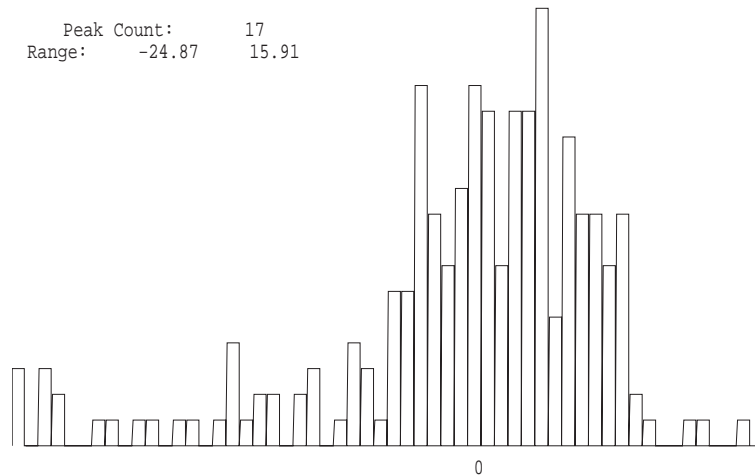


Figure 13.4 Histogram of residuals

where $\tilde{\beta}_D$ is the dense portion of β and C^{DD} is the dense portion of C^{-1} . This is the first 20 rows of `nin89.vrb`. Note that the first element is the estimated error variance, that is, 48.6802, see the variance component estimates in the `.asr` output.

```
0.486802E + 02    0.000000E + 00    0.000000E + 00    0.298660E + 01    0.000000E + 00
0.807551E + 01    0.470711E + 01    0.000000E + 00    0.456648E + 01    0.886687E + 01
-0.313123E + 00    0.000000E + 00    0.410031E + 01    0.476546E + 01    0.876708E + 01
0.295404E + 01    0.000000E + 00    0.343331E + 01    0.389620E + 01    0.416124E + 01
0.743616E + 01    0.163302E + 01    0.000000E + 00    0.377176E + 01    0.428109E + 01
0.472519E + 01    0.402696E + 01    0.837281E + 01    0.129013E + 01    0.000000E + 00
0.330076E + 01    0.347471E + 01    0.357605E + 01    0.316915E + 01    0.412130E + 01
0.768275E + 01    0.310018E + 00    0.000000E + 00    0.376637E + 01    0.419780E + 01
0.395693E + 01    0.383429E + 01    0.458492E + 01    0.378585E + 01    0.985202E + 01
0.226478E + 01    0.000000E + 00    0.379286E + 01    0.442457E + 01    0.439485E + 01
0.402503E + 01    0.440539E + 01    0.362391E + 01    0.502071E + 01    0.901191E + 01
0.508553E + 01    0.000000E + 00    0.393626E + 01    0.430512E + 01    0.423753E + 01
0.428826E + 01    0.417864E + 01    0.363341E + 01    0.444776E + 01    0.527289E + 01
0.855241E + 01    0.243687E + 01    0.000000E + 00    0.351386E + 01    0.369983E + 01
0.384055E + 01    0.330171E + 01    0.362019E + 01    0.352370E + 01    0.359516E + 01
0.392097E + 01    0.406762E + 01    0.801579E + 01    0.475935E + 01    0.000000E + 00
...

```

The first 5 rows of the lower triangular matrix in this case are

$$\begin{bmatrix} 48.6802 & & & & & & & \\ & 0 & 0 & & & & & \\ & 2.98660 & 0 & 8.07551 & & & & \\ & 4.70711 & 0 & 4.56648 & 8.86687 & & & \\ -0.313123 & 0 & 4.10031 & 4.76546 & 8.76708 & & & \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \end{bmatrix}$$

The `.vvp` file

The `.vvp` file contains the inverse of the average information matrix on the components scale. The file is formatted for reading back under the control of the `.pin` file described in Chapter 11. The matrix is lower triangular row-wise in the order the parameters are printed in the `.asr` file. This is `nin89.vvp` with the parameter estimates in the order error variance, spatial row correlation, spatial column correlation.

```
Variance of Variance components      3
  51.0852
  0.217089      0.318058E-02
  0.677748E-01 -0.201181E-02  0.649355E-02

```

The .rsv file

The `.rsv` file contains the variance parameters from the most recent iteration of a model. The primary use of the `.rsv` file is to supply the values for the `!CONTINUE` qualifier (see Table 5.4) and the `C` command line option (see Table 12.1). It contains sufficient information to match terms so that it can be used when the variance model has been changed. This is `nin89.rsv`.

```

76 6
      0.000000      0.000000      0.000000      1.000000      0.6553855      0.4375849
RSTRUCTURE              1  2
VARIANCE                 1  1  0 1.00000
STRUCTURE                22  1 10.655385
STRUCTURE                11  1 10.437585

```

The .dpr file

The `.dpr` file contains the data and residuals from the analysis in double precision binary form. The file is produced when the `!RES` qualifier (Table 4.3) is invoked. The file could be renamed with filename extension `.dbl` and used for input to another run of `ASReml`. Alternatively, it could be used by another Fortran program or package. Factors will have level codes if they were coded using `!A` or `!I`. All the data from the run plus an extra column of residuals is in the file. Records omitted from the analysis are omitted from the file.

The .pvc file

The `.pvc` file contains functions of the variance components produced by running a `.pin` file on the results of an `ASReml` run as described in Chapter 11. The `.pin` and `.pvc` files for a half-sib analysis of the Coopworth data are presented in Section 15.10.

The .pvs file

The `.pvs` file contains the predicted values formed when a `predict` statement is included in the job. Below is an edited version of `nin89.pvs`. See Section 3.6 for the `.pvs` file for the simple RCB analysis of the NIN data considered in that chapter.

```

title line      nin alliance trial      10 Sep 2002 04:20:15
                                     nin89

Ecode is E for Estimable, * for Not Estimable

```

Warning: mv_estimates is ignored for prediction

----- 1 -----
 Predicted values of yield

	variety	Predicted_Value	Standard_Error	Ecode	
predicted variety means	LANCER	24.0875	2.4645	E	
	BRULE	27.0749	2.4944	E	
	REDLAND	28.7949	2.5064	E	
	CODY	23.7752	2.4970	E	
	ARAPAHOE	27.0416	2.4417	E	
	NE83404	25.7212	2.4424	E	
	NE83406	25.3776	2.5028	E	
	NE83407	24.3978	2.6882	E	
	CENTURA	26.3525	2.4763	E	
	SCOUT66	29.1732	2.4361	E	
	:				
		NE87615	25.1238	2.4434	E
		NE87619	30.0267	2.4666	E
	NE87627	19.7126	2.4833	E	
SED summary	SED: Overall Standard Error of Difference		2.925		

The .tab file

The .tab file contains the simple variety means and cell frequencies. Below is an edited version of nin89.tab.

```
nin alliance trial
```

```
10 Sep 2002 04:20:15
```

```
Simple tabulation of yield
```

```
variety
LANCER          28.56    4
BRULE           26.07    4
REDLAND        30.50    4
CODY            21.21    4
ARAPAHOE       29.44    4
NE83404        27.39    4
NE83406        24.28    4
NE83407        22.69    4
CENTURA       21.65    4
SCOUT66        27.52    4
COLT           27.00    4
:
NE87615        25.69    4
NE87619        31.26    4
NE87627        23.23    4
```


13.5 ASReml output objects and where to find them

Table 13.2 presents a list of objects produced with each ASReml run and where to find them in the output files.

Table 13.2: ASReml output objects and where to find them

output object	found in	comment
analysis of variance table	<code>.asr</code> file	the analysis of variance table contains F-statistics for each term in the <i>fixed</i> part of the model. These provide for an incremental test of significance (see Section 6.10).
data summary	<code>.asr</code> file	includes the number of records read and retained for analysis, the minimum, mean, maximum, number of zeros, number of missing values per data field, factor/variable field distinction
elapsed time	<code>.asr</code> file	this can be determined by comparing the start time with the finishing time.
fixed and random effects	<code>.sln</code> file	the effects that were included in the dense portion of the solution are also printed in the <code>.asr</code> file with their standard error, a t-statistic for testing that effect and a t-statistic for testing it against the preceding effect in that factor.
heritability	<code>.pvc</code> file	placed in the <code>.pvc</code> file when postprocessing with a <code>.pin</code> file
histogram of residuals	<code>.res</code> file	and graphics file
intermediate results	<code>.asl</code> file	given if the <code>-DL</code> command line option is used.
mean/variance relationship	<code>.res</code> file	for non-spatial analyses ASReml prints the slope of the regression of <code>log(abs(residual))</code> against <code>log(predicted value)</code> . This regression is expected to be near zero if the variance is independent of the mean. A power of the mean data transformation might be indicated otherwise. The suggested power is approximately $(1-b)$ where b is the slope. A slope of 1 suggests a log transformation. This is indicative only and should not be blindly applied. Weighted analysis or identifying the cause of the heterogeneity should also be considered. This statistic is not reliable in genetic animal models or when <code>units</code> is included in the linear model because then the predicted value includes some of the residual.

Table of output objects and where to find them ASReml

output object	found in	comment
observed variance/covariance matrix formed from BLUPs and residuals	.res file	for an interaction fitted as random effects, when the first [outer] dimension is smaller than the inner dimension less 10, ASReml prints an observed variance matrix calculated from the BLUPs. The observed correlations are printed in the upper triangle. Since this matrix is not well scaled as an estimate of the underlying variance component matrix, a rescaled version is also printed, scaled according to the fitted variance parameters. The primary purpose for this output is to provide reasonable starting values for fitting more complex variance structure. The correlations may also be of interest. After a multivariate analysis, a similar matrix is also provided, calculated from the residuals.
phenotypic variance	.pvc file	placed in the .pvc file when postprocessing with a .pin file
plot of residuals against field position	graphics file	
possible outliers	.res file	these are residuals that are more than 3.5 standard deviations in magnitude
predicted (fitted) values at the data points	.yht file	these in the are printed in the second column
predicted values	.pvs file	given if a <code>predict</code> statement is supplied in the .as file.
REML log-likelihood	.asr file	the REML log-likelihood is given for each iteration. The REML log-likelihood should have converged
residuals	.yht file	and in binary form in .dpr file; these are printed in column 3. Furthermore, for multivariate analyses the residuals will be in data order (traits within records). However, in a univariate analysis with missing values that are not fitted, there will be fewer residuals than data records - there will be no residual where the data was missing so this can make it difficult to line up the values unless you can manipulate them in another program (spreadsheet).
score	.asl file	given if the <code>-DL</code> command line option is used.

Table of output objects and where to find them ASReml

output object	found in	comment
tables of means	.tab file .pvs file	simple averages of cross classified data are produced by the tabulate directive to the .tab file. Adjusted means predicted from the fitted model are written to the .pvs file by the predict directive.
variance of variance parameters	.vvp file	based on the inverse of the average information matrix
variance parameters	.asr file .res file	the values at each iteration are printed in the .res file. The final values are arranged in a table, printed with labels and converted if necessary to variances.
variogram	graphics file	

14

Error messages

Introduction

Common problems

Things to check in the `.asr` file

An example

Error messages

Warning messages

14.1 Introduction

When ASReml finds an inconsistency it prints an error message to the screen (or the .asl file) and dumps the current information to the .asr file. Below is the screen output for a job that has been terminated due to an error. If a job has an error you should

- try and identify the problem from the error message in the Fault: line and the text of the Last line read: (this appears twice in the file to make it easier to find),
- check that all labels have been defined and are in the correct case,
- some errors arise from conflicting information; the error may point to something that appears valid but is inconsistent with something that is wrong earlier in the file,
- reduce to a simpler model and gradually build up to the desired analysis - this should help to identify the exact location of the problem.

If the problem is not resolved from the above list, you may need to email the Help Desk at asreml@vsn-intl.com. Please send the .as file, (a sample of) the data, the .asr file and the .asl file produced by running

```
asreml -dl basename.as
```

See Chapter 12 The -dl command line option invokes debug mode and sends all non-graphics screen output to the .asl file.

In this chapter we show some of the common typographical problems. Errors arising from attempts to fit an inappropriate model are often harder to resolve. Following is an example of output when the data file is not correctly named or is not present (ASReml tries to interpret the filename as a variable name when the file is not found).

```

ASReml 1.0 [31 Aug 2002]  nin alliance trial
memory info 10 Sep 2002 05:13:15.253  8.00 Mbyte  Windows  ninerr1
Licensed to: Arthur Gilmour
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support  *
* Use !{ and !} when fitting covariance across terms      *
***** ARG *
working direc- Folder: C:\asr\ex\manex
tory
```

```

Warning: FIELD DEFINITION lines should be INDENTED
messages      There is no file called nine.asd
Invalid label for data field: 'nine.asd' contains a reserved character
              or may get confused with a previous label or reserved word
Fault 12 FORMAT error reading data structures
Last line read was:  nine.asd !slip 1
Currently defined structures, COLS and LEVELS
  1 variety          -1   56   0   0   0   0
  2 id               -2    1   0   0   0   0
  3 pid              -3    1   0   0   0   0
  4 raw              -4    1   0   0   0   0
  5 repl             -5    4   0   0   0   0
  6 nloc             -6    1   0   0   0   0
  7 yield            -7    1   0   0   0   0
  8 lat              -8    1   0   0   0   0
  9 long             -9    1   0   0   0   0
10 row              -10   22   0   0   0   0
11 column            -11   11   0   0   0   0
filename! 12 nine.asd    0    0   0   0   0   0
ninerr1 C:\asr\ex\manex
      12 factors defined [max 500].
      0 variance parameters [max 900].  2 special structures
last line read Last line read was:  nine.asd !slip 1
              12   12   -1    0  1000
fault message Finished: 10 Sep 2002 05:13:16.745  FORMAT error reading data structures  :
```

14.2 Common problems

Common problems in coding ASReml are as follows:

- a variable name has been misspelt; variable names are case sensitive,
- a model term has been misspelt; model term functions and reserved words (`mu`, `Trait`, `mv`, `units`) are case sensitive,
- the data file name is misspelt or the wrong path has been given - the pathname may not include embedded blanks,
- a qualifier has been misspelt or is in the wrong place,
- there is an inconsistency between the variance header line and the structure definition lines presented,

- failure to use commas appropriately in model definition lines,
- there is an error in the R structure definition lines,
- there is an error in the G structure definition lines,
 - there is a factor name error,
 - there is a missing parameter,
- there is an error in the predict statement ,
- model term `mv` not included in the model when there are missing values in the data when the user wishes to retain their records in the analysis.

The most common problem in running ASReml is that a variable label is misspelt.

14.3 Things to check in the `.asr` file

The information that ASReml dumps in the `.asr` file when an error is encountered is intended to give you some idea of the particular error:

- if there is no data summary ASReml has failed before or while reading the model line,
- if ASReml has completed one iteration the problem is probably associated with starting values of the variance parameters or the logic of the model rather than the syntax *per se*.

Part of the file `nin89.asr` presented in Chapter 13 is displayed below to indicate the lines of the `.asr` file that should be checked. You should check that

- sufficient workspace has been obtained,
- the records read/lines read/records used are correct,
- mean min max information is correct for each variable,
- ANOVA table has the expected degrees of freedom.

```

workspace  ASReml 1.0 [31 Aug 2002]  nin alliance trial
           10 Sep 2002 04:20:15.350  \textcolor{blue} 8.00 Mbyte \textcolor{black} Windows nin89
           Licensed to: Arthur Gilmour
           * ASReml 2002 *****
           * Contact asreml@vsn-intl.com for licensing and support  *
           * Use !{ and !} when fitting covariance across terms    *

```

```

***** ARG *
working direc- Folder: C:\asr\ex\manex
tory          QUALIFIERS: !SKIP 1
              Reading nin.asd FREE FORMAT skipping 1 lines
              Univariate analysis of yield
records read   Using 242 records of 242 read
data summary  Model term      Size Type  COL  Minimum  Mean  Maximum  #zero #miss
              1 variety      56 Factor  1     1      26.4545  56       0     0
              2 id           2         2  1.000    26.45   56.00    0     0
              3 pid          3         3 1101.    2628.   4156.    0    18
              4 raw          4         4 21.00    510.5   840.0    0    18
              5 repl        4 Factor  5     1      2.4132   4       0     0
              6 nloc        6         6 4.000    4.000   4.000    0     0
              7 yield       1 Variate 7  1.050    25.53   42.00    0    18
              8 lat         8         8 4.300    25.80   47.30    0     0
              9 long        9         9 1.200    13.80   26.40    0     0
             10 row         22 Factor 10     1     11.5000  22      0     0
             11 column     11 Factor 11     1      6.0000  11      0     0
             12 mu          1 Constant Term
             13 mv_estimates 18 Missing value
                22 AR=AutoReg 0.3000
                11 AR=AutoReg 0.3000
              Forming 75 equations: 57 dense.
              Initial updates will be shrunk by factor 0.316
              NOTICE: 1 (more) singularities,
iteration     1 LogL=-419.949 S2= 39.387 168 df 1.000 0.3000 0.3000
sequence     2 LogL=-410.420 S2= 38.350 168 df 1.000 0.4030 0.3461
              3 LogL=-402.549 S2= 40.253 168 df 1.000 0.5202 0.3999
              4 LogL=-399.465 S2= 46.228 168 df 1.000 0.6233 0.4423
              5 LogL=-399.331 S2= 48.139 168 df 1.000 0.6475 0.4414
              6 LogL=-399.324 S2= 48.580 168 df 1.000 0.6535 0.4389
              7 LogL=-399.324 S2= 48.680 168 df 1.000 0.6550 0.4379
              8 LogL=-399.324 S2= 48.705 168 df 1.000 0.6554 0.4376
              Final parameter values 1.0000 0.65549 0.43750

Source      Model terms  Gamma  Component  Comp/SE  % C
Variance    242 168 1.00000 48.7045 6.81 0 P
Residual    AR=AutoR 22 0.655488 0.655488 11.63 0 U
Residual    AR=AutoR 11 0.437497 0.437497 5.43 0 U

```


ANOVA

```

Analysis of Variance
12 mu
1 variety
NOTICE: ASREml cannot yet determine the correct denominator DFs for
these F statistics. They will not exceed the Residual DF
which applies only if all components are KNOWN.

Estimate      Standard Error    T-value    T-prev
1 variety
BRULE          2.98387          2.84137          1.05
REDLAND        4.70621          2.97738          1.58    0.62
CODY          -0.316046        2.96066          -0.11   -1.76
:              :                  :           :
NE87615        1.03251          2.93379          0.35   -0.40
NE87619        5.93687          2.84941          2.08    1.82
NE87627       -4.37848          2.99740          -1.46   -3.72
12 mu
57    24.0892          2.46480          9.77
13 mv_estimates      18 effects fitted
6 possible outliers: see .res file
Finished: 10 Sep 2002 04:20:34.508  LogL Converged

```

outliers?

14.4 An example

See 2a in Section 7.3

This is the command file for a simple RCB analysis of the NIN variety trial data in the first part. However, this file contains eight common mistakes in coding ASREml. We also show two common mistakes associated with spatial analyses in the second part. The errors are highlighted and the numbers indicate the order in which they are detected. Each error is discussed with reference to the output written to the `.asr` file. A summary of the errors is as follows:

1. data file not found,
2. unrecognised qualifier,
3. incorrectly defined alphanumeric factor,

```

nin alliance trial
variety 56 # 3.
id
pid
raw
repl 4
nloc
yield
lat
long
row 22
column 11
nin.asd !slip 1 !dopart $1
# 1. & 2.
!part 1
yield~mu variety # 4.
!r Repl # 5.
0 0 1
Repl 1 # 6.
2 0 IDV 0.1 # 7.
!part 2
yield~mu variety # 9.
1 2
11 row AR1 .1 #10.
22 col AR1 .1
!part
predict voriety # 8.

```

4. comma missing from first line of model,
5. misspelt variable label in linear model,
6. misspelt variable label in G structure header line,
7. wrong levels declared in G structure model line,
8. misspelt variable label in predict statement.
9. mv omitted from spatial model
10. wrong levels declared in R structure model lines

1. Data file not found

Running this job produces the `.asr` file in Section 14.1. The first problem is that ASReml cannot find the data file `nine.asd` as indicated in the error message above the **Fault** line. ASReml reports the last line read before the job was terminated, an error message **FORMAT error reading data structures**

```
nin alliance trial
:
nine.asd !slip 1
yield ~ mu variety
:
```

and other information obtained to that point. In this case the program only made it to the data file definition line in the command file. Since `nine.asd` commences in column 1, ASReml checks for a file of this name (in the working directory since no path is supplied). Since ASReml did not find the data file it tried to interpret the line as a variable definition but `.` is not permitted in a variable label. The problem is either that the filename is misspelt or a pathname is required. In this case the data file was given as `nine.asd` rather than `nin.asd`.

2. An unrecognised qualifier and 3. An incorrectly defined factor

After supplying the correct pathname and re-running the job, ASReml produces the warning message

```
WARNING: Unrecognised qualifier at character 22 !slip 1
```

followed by the fault message

```
ERROR Reading the data.
```

The warning does not cause the job to terminate immediately but arises because

!slip is not a recognised data file line qualifier; the correct qualifier is !skip. The job terminates when reading the header line of the nin.asd file which is alphabetic when it is expecting numeric values. The following output displays the error message produced.

```

ASReml 1.0 [31 Aug 2002]  nin alliance trial
10 Sep 2002 05:13:58.815    8.00 Mbyte  Windows  ninerr2
Licensed to: Arthur Gilmour
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support  *
* Use !{ and !} when fitting covariance across terms      *
***** ARG *
Folder: C:\asr\ex\manex
QUALIFIERS: !SLIP 1
Warning: Unrecognised qualifier at character  9 !SLIP 1
Reading nin.asd  FREE FORMAT skipping    0 lines
Univariate analysis of yield
Field  1 [variety] of record      1 [line    1] is not valid.
Increase the number of levels in factor variety
Fault    0  ERROR Reading the data
Last line read was:  variety id pid raw rep nloc yield lat long row column
Currently defined structures, COLS and LEVELS
   1 variety                                1   56   56   0   0   0
   :
  12 mu                                    0    1   -8   0  -1   0
ninerr2 nin.asd
Model specification:  TERM LEVELS GAMMAS
mu                    0    0.000
variety               0    0.000
  12 factors defined [max 500].
  0 variance parameters [max 900].  2 special structures
Last line read was:  variety id pid raw rep nloc yield lat long row column
  12    0    0    0  1000
Finished: 10 Sep 2002 05:14:00.348  ERROR Reading the data

```

error
hint

Fixing the error by changing !slip to !skip however still produces the fault message

ERROR Reading the data.

The portion of output given below shows that ASReml has baulked at the name

LANCER in the first field on the first data line. This alphabetic data field is not declared as alphabetic. The correct data field definition for variety is

```
variety !A
```

to indicate that variety is a character field.

```
ASReml 1.0 [31 Aug 2002] nin alliance trial
10 Sep 2002 05:14:00.818 8.00 Mbyte Windows ninerr3
Licensed to: Arthur Gilmour
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support *
* Use !{ and !} when fitting covariance across terms *
***** ARG *
Folder: C:\asr\ex\manex
QUALIFIERS: !SKIP 1
Reading nin.asd FREE FORMAT skipping 1 lines
Univariate analysis of yield
Field 1 [LANCER] of record 1 [line 1] is not valid.
Increase the number of levels in factor variety
Fault 0 ERROR Reading the data
Last line read was: LANCER 1 NA NA 1 4 NA 4.3 1.2 1 1
Currently defined structures, COLS and LEVELS
  1 variety 1 56 56 0 0 0
  :
12 mu 0 1 -8 0 -1 0
ninerr3 variety id pid raw rep nloc yield lat
Model specification: TERM LEVELS GAMMAS
mu 0 0.000
variety 0 0.000
12 factors defined [max 500].
0 variance parameters [max 900]. 2 special structures
Last line read was: LANCER 1 NA NA 1 4 NA 4.3 1.2 1 1
12 0 0 0 1000
Finished: 10 Sep 2002 05:14:02.000 ERROR Reading the data
```

4. A missing comma and 5. A misspelt factor name in linear model

The model has been written over two lines but ASReml does not realise this because the first line does not end with a comma.

```
nin alliance trial
variety !A
:
repl 4
:
nin.asd !skip 1
yield ~ mu variety
!r Repl
:
```

The missing comma causes the error message

Variance header: SECTIONS DIMNS GSTRUCT

as ASReml tries to interpret the second line of the model (see `Last line read`) as the variance header line. The `.asr` file is displayed below. Note that the data has now been successfully read as indicated by the data summary. You should always check the data summary to ensure that the correct number of records have been detected and the data values match the names appropriately.

```
ASReml 1.0 [31 Aug 2002]  nin alliance trial
10 Sep 2002 05:14:02.310   8.00 Mbyte  Windows  ninerr4
Licensed to: Arthur Gilmour
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support *
* Use !{ and !} when fitting covariance across terms *
***** ARG *
Folder: C:\asr\ex\manex
QUALIFIERS: !SKIP 1
Reading nin.asd  FREE FORMAT skipping    1 lines
Univariate analysis of yield
Using      224 records of    242 read
Model term      Size Type    COL  Minimum   Mean      Maximum  #zero #miss
  1 variety          56 Factor    1      1    28.5000      56      0     0
  2 id                2 1.000     28.50    56.00      0     0
  3 pid              3 1101.    2628.    4156.      0     0
  4 raw              4 21.00    510.5    840.0      0     0
  5 repl            4 Factor    5      1     2.5000      4      0     0
  6 nloc              6 4.000     4.000    4.000      0     0
  7 yield            1 Variate   7 1.050    25.53    42.00      0     0
  8 lat              8 4.300    27.22    47.30      0     0
  9 long              9 1.200    14.08    26.40      0     0
 10 row              22 Factor   10      1   11.7321     22      0     0
 11 column           11 Factor   11      1    6.3304     11      0     0
 12 mu                1 Constant Term
Fault      0 Variance header: SECTIONS DIMNS GSTRUCT
Last line read was:   !r Repl 0 0 0 0
ninerr4 variety id    pid    raw    rep    nloc    yield    lat
Model specification:  TERM LEVELS GAMMAS
```

```

variety                56    0.000
mu                     1    0.000
  12 factors defined [max 500].
    0 variance parameters [max 900].  2 special structures
Final parameter values
Last line read was:    !r Repl 0 0 0 0
    12    0    242  224  1000
Finished: 10 Sep 2002 05:14:03.552  Variance header: SECTIONS DIMNS GSTRUCT

```

Inserting a comma on the end of the first line of the model to give

```

yield ~ mu variety,
!r Repl

```

solves that problem but produces the error message

Error reading model factor list

because `Repl` should have been spelt `repl`. Portion of the output is displayed. Since the model line is parsed before the data is read, this run failed before reading the data.

```

ASReml 1.0 [31 Aug 2002]  nin alliance trial
10 Sep 2002 05:14:03.863  8.00 Mbyte  Windows  ninerr5
Licensed to: Arthur Gilmour
* ASReml 2002 *****
* Contact asreml@vsn-intl.com for licensing and support *
* Use !{ and !} when fitting covariance across terms *
***** ARG *
Folder: C:\asr\ex\manex
QUALIFIERS: !SKIP 1
Reading nin.asd  FREE FORMAT skipping    1 lines
Fault    0  Error reading model factor list
Last line read was:    Repl
Currently defined structures, COLS and LEVELS
  1 variety                1    2    2    0    0    0
:
12 mu                     0    1   -8    0   -1    0
ninerr5 variety id      pid    raw    rep    nloc    yield    lat
  12 factors defined [max 500].
    0 variance parameters [max 900].  2 special structures
Last line read was:    Repl

```

```

12      0      -1      0      1000
Finished: 10 Sep 2002 05:14:05.034   Error reading model factor list

```

6. Misspelt factor name and 7. Wrong levels declaration in the G structure definition lines

The next error message that ASReml presents is

G structure header: Factor, order

indicating that there is something wrong in the G structure definition lines. In this case the replicate term in the first G structure definition line has been spelt incorrectly. To correct this error replace Repl with repl.

```

nin alliance trial
:
nin.asd !skip 1
yield ~ mu variety
!r Repl
0 0 1
Repl 1
2 0 IDV 0.1

```

```

ASReml 1.0 [31 Aug 2002] nin alliance trial
10 Sep 2002 05:14:05.355   8.00 Mbyte Windows ninerr6
Licensed to: Arthur Gilmour
* ASReml 2002 *****
* Contact asrem1@vsn-intl.com for licensing and support *
* Use !{ and !} when fitting covariance across terms *
***** ARG *
Folder: C:\asr\ex\manex
QUALIFIERS: !SKIP 1
Reading nin.asd FREE FORMAT skipping 1 lines
Univariate analysis of yield
Using 224 records of 242 read

```

Model term	Size	Type	COL	Minimum	Mean	Maximum	#zero	#miss
1 variety	56	Factor	1	1	28.5000	56	0	0
2 id			2	1.000	28.50	56.00	0	0
3 pid			3	1101.	2628.	4156.	0	0
4 raw			4	21.00	510.5	840.0	0	0
5 repl	4	Factor	5	1	2.5000	4	0	0
6 nloc			6	4.000	4.000	4.000	0	0
7 yield	1	Variate	7	1.050	25.53	42.00	0	0
8 lat			8	4.300	27.22	47.30	0	0
9 long			9	1.200	14.08	26.40	0	0
10 row	22	Factor	10	1	11.7321	22	0	0
11 column	11	Factor	11	1	6.3304	11	0	0

```

12 mu          1 Constant Term
Fault      1 G structure header: Factor, order
Last line read was: Repl 1 0 0 0 0
ninerr6 variety id      pid      raw      rep      nloc      yield      lat
Model specification:  TERM LEVELS GAMMAS
variety                                56      0.000
mu                                      1      0.000
repl                                    4      0.100
SECTIONS      224      4      1
  TYPE        0      0      0
STRUCT      224      0      0      0      0      0      0
  12 factors defined [max 500].
  4 variance parameters [max 900].  2 special structures
Final parameter values                                0.10000      1.0000
Last line read was: Repl 1 0 0 0 0
  12      1      242  224  1000
Finished: 10 Sep 2002 05:14:06.556  G structure header: Factor, order

```

Fixing the header line, we then get the error message

Structure / Factor mismatch

This arose because repl has 4 levels but we have only declared 2 in the G structure model line. The G structure should read

```
repl 1
```

```
4 0 IDV 0.1
```

The last lines of the output with this error are displayed below.

```

11 column      11 Factor  11      1      6.3304      11      0      0
12 mu          1 Constant Term
  2 identity    0.1000
Structure for repl          has      2 levels defined
Fault      1 Structure / Factor mismatch
Last line read was: 2 0 IDV 0.1 0 0 0 0 0
ninerr7 variety id      pid      raw      rep      nloc      yield      lat
Model specification:  TERM LEVELS GAMMAS
variety                                56      0.000
mu                                      1      0.000
repl                                    4      0.100
SECTIONS      224      4      1
  TYPE        0      0      1002

```



```

STRUCT      224      0      0      0      0      0      0
           2      1      0      5      0      1      0
12 factors defined [max 500].
  5 variance parameters [max 900].  2 special structures
Final parameter values                0.10000    1.0000    0.10000

Last line read was:  2 0 IDV 0.1 0 0 0 0 0
 12   1   242  224  1000
Finished: 10 Sep 2002 05:14:08.069   Structure / Factor mismatch

```

8. A misspelt factor name in the predict statement

The final error in the job is that a factor name is misspelt in the predict statement. This is a non-fatal error. The faulty statement is simply ignored by ASReml and no .pvs file is produced. To rectify this statement correct `voriety` to `variety`.
see Chapter 13

9. Forgetting `mv` in a spatial analysis

The first error message from running part 2 of the job is

```
R structures imply 0 + 242 records:  only 224 exist
```

Checking the seventh line of the output below, we see that there were 242 records read but only 224 were retained for analysis. There are three reasons records are dropped.

1. the `!FILTER` qualifier has been specified,
2. the `!D` transformation qualifier has been specified and
3. there are missing values in the response variable and the user has not specified that they be estimated.

The last applies here so we must change the model line to read `yield ~ mu variety mv`.

```

:
Reading nin.asd  FREE FORMAT skipping    1 lines
Univariate analysis of yield
Using      224 records of    242 read
Model term      Size Type  COL  Minimum   Mean      Maximum  #zero #miss
  1 variety      56 Factor   1    1   28.5000     56      0    0
:
12 mu            1 Constant Term
 11 AR=AutoReg   0.1000

```

```

      22 AR=AutoReg      0.1000
Maybe you need to include 'mv' in the model
Fault      1 R structures imply      0 242 records: only 224 e+
Last line read was: 22 column AR1 0.1 0 0 0 0 0
ninerr9 variety id      pid      raw      rep      nloc      yield      lat
Model specification: TERM LEVELS GAMMAS
variety                                56      0.000
mu                                     1      0.000
SECTIONS      242      3      1
STRUCT        11      1      1      4      0      1      10
              22      1      1      5      0      1      11
12 factors defined [max 500].
5 variance parameters [max 900]. 2 special structures
Final parameter values                0.0000  -.10000E-360.10000
0.10000
Last line read was: 22 column AR1 0.1 0 0 0 0 0
12      1      242      224      1000
Finished: 10 Sep 2002 05:14:11.143 R structures imply      0 242 records: only
224 exist +

```

10. Field layout error in a spatial analysis

The final common error we highlight is the misspecification of the field layout. In this case we have 'accidentally' switched the levels in rows and columns. However, ASReml can detect this error because we have also asked it to sort the data into field order. Had sorting not been requested, ASReml would not have been able to detect that the lines of the data file were not sorted into the appropriate field order and spatial analysis would be wrong.

```

:
Reading nin.asd FREE FORMAT skipping      1 lines
Univariate analysis of yield
Using      242 records of      242 read
Model term      Size Type      COL      Minimum      Mean      Maximum      #zero      #miss
1 variety      56 Factor      1      1      26.4545      56      0      0
:
10 row      22 Factor      10      1      11.5000      22      0      0
11 column      11 Factor      11      1      6.0000      11      0      0
12 mu      1 Constant Term
13 mv_estimates 18 Missing value

```

```

    11 AR=AutoReg    0.1000
    22 AR=AutoReg    0.1000
Warning: Spatial mapping information for side 1 of order 11
        ranges from    1.0 to    22.0
Warning: Spatial mapping information for side 2 of order 22
        ranges from    1.0 to    11.0
Fault    2 Sorting data into field order
Last line read was: 22 column AR1 0.1 0 0 0 0 0
ninerr10 variety id    pid    raw    rep    nloc    yield    lat
Model specification:  TERM LEVELS GAMMAS
variety                                56    0.000
mu                                      1    0.000
mv_estimates                           18    0.000
SECTIONS    242    4    1
STRUCT      11    1    1    5    0    1    10
           22    1    1    6    0    1    11
    13 factors defined [max 500].
    6 variance parameters [max 900].    2 special structures
Final parameter values                0.0000    -.10000E-360.10000
0.10000
Last line read was: 22 column AR1 0.1 0 0 0 0 0
    13    2    242    242    1000
Finished: 10 Sep 2002 05:14:12.655    Sorting data into field order

```

14.5 Information, Warning and Error messages

ASReml prints information, warning and error messages in the `.asr` file. The major information messages are in Table 14.2. A list of warning messages together with the likely meaning(s) is presented in Table 14.1. Error messages with their probable cause(s) is presented in Table 14.3.

Table 14.1: Some information messages and comments

information message	comment
<code>Logl converged</code>	the REML log-likelihood changes between iterations were less than $0.002 * \text{iteration number}$.
<code>Logl converged, parameters not converged</code>	the change in REML log-likelihood was small and convergence was assumed but the parameters are, in fact, still changing.
<code>Logl not converged</code>	the maximum number of iterations was reached before the REML log-likelihood converged. Examine the sequence of estimates. You may need more iterations in which case restart with <code>C</code> command line option (see Section 12.3 on job control). Otherwise restart with more appropriate initial variances. It may be necessary to simplify the model and estimate the dominant components before estimating other terms.

Table 14.2: List of warning messages and likely meaning(s)

warning message	likely meaning
Warning: <i>e</i> missing values generated by !^ transformation	data values should be positive.
Warning: <i>i</i> singularities in AI matrix	usually means variance model is overparameterized.
Warning: <i>m</i> variance structures were modified	the structures are probably at the boundary of the parameter space.
Warning: <i>n</i> missing values were detected in the design	either use !MVINCLUDE or delete the records.
Warning: <i>n</i> negative weights	it is better to avoid negative weights unless you can check ASReml is doing the correct thing with them.
Warning: <i>r</i> records were read from multiple lines	check you have the intended number of fields per line.
WARNING <i>term</i> has more levels [##] than expected [##]:	You have probably mis-specified the number of levels in the factor or omitted the !I qualifier (see Section 5.4 on data field definition syntax). ASReml corrects the number of levels.
Warning: <i>term</i> in the predict !IGNORE list	the term did not appear in the model.
Warning: <i>term</i> in the predict !USE list	the term did not appear in the model.
Warning: <i>term</i> is ignored for prediction	terms like <code>units</code> and <code>mv</code> cannot be included in prediction.
Warning: Check if you need the !RECODE qualifier	!RECODE may be needed if the binary file was not prepared with ASReml.
Warning: Code B - fixed at a boundary (!GP)	suggest drop the term and refit the model.
Warning: Eigen analysis check of US matrix skipped	matrix is probably OK.
WARNING: Extra lines on the end of the input file ...:	this indicates that there are some lines on the end of the .as file that were not used. The first "extra" line is displayed. This is only a problem if you intended ASReml to read these lines.
Warning: Fewer levels found in <i>term</i>	ASReml increases to the correct value.
Warning: FIELD DEFINITION lines should be INDENTED	indent them to avert this message.

List of warning messages and likely meaning(s)

warning message	likely meaning
Warning: Fixed levels for factor	user nominated more levels than are permitted.
Warning: Initial gamma value is zero	constraint parameter is probably wrongly assigned.
Warning: Invalid argument.	fix the argument.
Warning: It is usual to include Trait	in multivariate analysis model.
Warning: LogL Converged; Parameters Not Converged	you may need more iterations.
Warning: LogL not converged	restart to do more iterations (see !CONTINUE).
Warning: Missing cells in table	missing cells are normally not reported.
Warning: More levels found in term	consider setting levels correctly.
Warning: PREDICT LINE IGNORED - TOO MANY	limit is 100.
Warning: PREDICT statement is being ignored	because it contains errors.
Warning: Second occurrence of term dropped	if you really want to fit this term twice, create a copy with another name.
Warning: Spatial mapping information for side	gives details so you can check ASReml is doing what you intend.
Warning: Standard errors	that is, these standard errors are approximate.
Warning: SYNTAX CHANGE: text may be invalid	use the correct syntax.
Warning: The !A qualifier ignored when reading BINARY data	the !A fields will be treated as factors but are not encoded.
Warning: The !SPLINE qualifier has been redefined.	use correct syntax.
Warning: The !X !Y !G qualifiers are ignored. There is no data to plot	revise the qualifier arguments.
Warning: The estimation was ABORTED	do not accept the estimates printed.
Warning: The labels for predictions are erroneous	the labels for predicted terms are probably out of kilter. Try a simpler predict statement. If the problem persists, send for help.

List of warning messages and likely meaning(s)

warning message	likely meaning
Warning: This US structure is not positive definite	check the initial values.
Warning: Unrecognised qualifier at character	the qualifier either is misspelt or is in the wrong place.
Warning: US matrix was not positive definite: MODIFIED	the initial values were modified.
Warning: User specified spline points	the points have been rescaled to suit the data values.
Warning: Variance parameters were modified by BENDING	ASReml may not have converged to the best estimate.
WARNING Likelihood decreased. Check gammas and singularities.:	a common reason is that some constraints have restricted the gammas. Add the !GU qualifier to any factor definition whose gamma value is approaching zero (or the correlation is approaching (-)1. Alternatively, more singularities may have been detected. You should identify where the singularities are expected and modify the data so that they are omitted or consistently detected. One possibility is to centre and scale covariates involved in interactions so that their standard deviation is close to 1.

Table 14.3: Alphabetical list of error messages and probable cause(s)

error message	probable cause
Convergence failure:	<p>the program did not proceed to convergence because the REML log-likelihood was fluctuating wildly. One possible reason is that some singular terms in the model are not being detected consistently. Otherwise, the updated G structures are not positive definite. There are some things to try:</p> <ul style="list-style-type: none"> - define US structures as positive definite by using !GP, - supply better starting values, - fix parameters that you are confident of while getting better estimates for others (that is, fix variances when estimating covariances), - fit a simpler model, - reorganise the model to reduce covariance terms (for example, use CORUH instead of US.)
Error reading the model factor list:	the model specification line is in error: a variable is probably misnamed.
Error reading the data:	the data file could not be interpreted: alphanumeric fields need the !A qualifier.
Error reading the data file name:	data file name may be wrong
Error opening the file:	the named file did not exist or was of the wrong file type (binary = unformatted, sequential).
Error structures are wrong size:	the declared size of the error structures does not match the actual number of data records.
Error in R structure:	the error model is not correctly specified.
FORMAT error reading factor Definitions:	<p>Likely causes are</p> <ul style="list-style-type: none"> - bad syntax or invalid characters in the variable labels; variable labels must not include any of these symbols; ! -+(:#\$ and ., - the data file name is misspelt, - there are too many variables declared or there is no valid <i>value</i> supplied with an arithmetic transformation option.
Getting Pedigree:	an error occurred processing the pedigree. The pedigree file must be ascii, free format with ANIMAL, SIRE and DAM as the first three fields.

Alphabetical list of error messages and probable cause(s)

error message	probable cause
G-structure header: Factor order:	there is a problem reading G structure header line. The line must contain the name of a term in the linear model spelt exactly as it appears in the model
G structure: ORDER 0 MODEL GAMMAS:	a G structure line cannot be interpreted.
Insufficient knots for :	there must be at least 3 distinct data values
Invalid factor in model:	a term in the <i>model specification</i> is not among the terms that have been defined. Check the spelling.
Invalid model factor ... :	there is a problem with the named variable.
Invalid definition of a factor:	there is a problem with forming one of the <i>generated</i> factors. The most probable cause is that an interaction cannot be formed.
Invalid weight/filter column number:	the weight and filter columns must be data fields. Check the data summary.
Invalid factor definition:	a term in the model has no levels
Missing values not allowed here:	missing observations have been dropped so that multivariate structure is messed up.
Maximum number of special structures exceeded	special structures are weights, the Ainverse and GIV structures. The limit is 8 and so no more than 6 GIV structures can be defined.
Negative Sum of Squares:	This is typically caused by negative variance parameters; try changing the starting values or using the !STEP option. If the problem occurs after several iterations it is likely that the variance components are very small. Try simplifying the model. In multivariate analyses it arises if the error variance is (becomes) negative definite. Try specifying !GP on the structure line for the error variance.
NFACT out of range:	too many terms are being defined.
Increase workspace with -Ss command line option:	on some systems the command line option -S3 may help. If the data set is not extremely big, check the data summary.
Out of memory absorbing:	use -S option to increase the memory allocation. It may be possible to revise the models to increase sparsity.

Alphabetical list of error messages and probable cause(s)

error message	probable cause
Out of memory: forming design:	factors are probably not declared properly. Check the number of levels. Possibly use the <code>-S</code> command line option.
Overflow structure table:	occurs when space allocated for the structure table is exceeded. There is room for three structures for each model term for which G structures are explicitly declared. The error might occur when ASReml needs to construct rows of the table for structured terms when the user has not formally declared the structures. Increasing <i>g</i> on the variance header line for the number of <i>G structures</i> (see page 89) will increase the space allocated for the table. You will need to add extra explicit declarations also.
No residual variation:	after fitting the model, the residual variation is essentially zero, that is, the model fully explains the data. If this is intended, use the <code>!BLUP 1</code> qualifier so that you can see the estimates. Otherwise check that the dependent values are what you intend and then identify which variables explain it. Again, the <code>!BLUP 1</code> qualifier might help.
PROGRAM failed in AIABS:	increase memory.
Pedigree coding errors:	check the pedigree file and see any messages in the output. Check that identifiers and pedigrees are in chronological order.
Pedigree factor has wrong size:	the A-inverse factors are not the same size as the A-inverse. Delete the <code>ainverse.bin</code> file and rerun the job.
Pedigree too big:	program will need to be recompiled if file is correct.
Reading factor names:	something is wrong in the terms definitions. It could also be that the data file is misnamed.
Reading the data:	if you read less data than you expect, there are two likely explanations. First, the data file has less fields than implied by the data structure definitions (you will probably read half the expected number). Second, there is an alphanumeric field where a numeric field is expected.

Alphabetical list of error messages and probable cause(s)

error message	probable cause
Reading Update step size:	check the !STEP qualifier argument.
Variance header: SEC DIM GSTRUCT:	error with the variance header line. Often, some other error has meant that the wrong line is being interpreted as the variance header line. Commonly, the model is written over several lines but the incomplete lines do not all end with a comma.
R structure error ORDER SORTCOL MODEL GAMMAS:	an error reading the error model.
Segmentation fault:	this is a Unix memory error. The first thing to try is to increase the memory workspace using the -S (see Section 12.3 on memory) command line option. Otherwise you may need to send your data and the .as files to the Help Desk for debugging.
Sorting the data into field order:	the field order coding in the spatial error model does not generate a complete grid with one observation in each cell; missing values may be deleted: they should be fitted. Also may be due to incorrect specification of number of rows or columns.
STOP SCRATCH FILE DATA STORAGE ERROR:	ASReml attempts to hold the data on a scratch file. Check that the disk partition where the scratch files might be written is not too full; use the !NOSCRATCH qualifier to avoid these scratch files.
Structure/ Factor mismatch:	the declared size of a variance structure does not match the size of the model term that it is associated with.
Too BIG! Use -S option:	memory overflow. Simplify the model, check factor definitions and increase memory using the -S command line option (see Section 12.3 on memory).
Too many alphanumeric factor level labels:	if the factor level labels are actually all integers, use the !I option instead. Otherwise, you will have to convert a factor with alphanumeric labels to numeric sequential codes external to ASReml so that an !A option can be avoided.

Alphabetical list of error messages and probable cause(s)

error message	probable cause
Unable to invert R or G [US?] matrix:	this message occurs when there is an error forming the inverse of a variance structure. The probable cause is a non positive definite (initial) variance structure (US, CHOL and ANTE models). It may also occur if an <i>identity by unstructured</i> (ID \otimes US) error variance model is not specified in a multivariate analysis (including !ASMV), see Chapter 8. If the failure is on the first iteration, the problem is with the starting values. If on a subsequent iteration, the updates have caused the problem. You could try reducing the updates by using the !STEP qualifier. Otherwise, you could try fitting an alternative parameterisation. The CORGH model may be more stable than the US model.
Unable to invert R or G [CORR?] matrix:	generally refers to a problem setting up the mixed model equations. Most commonly, it is caused by a non positive definite matrix.

Introduction

Split plot design - Oats

Unbalanced nested design - Rats

Source of variability in unbalanced data - Volts

Balanced repeated measures - Height

Spatial analysis of a field experiment - Barley

Unreplicated early generation variety trial - Wheat

Paired Case-Control study - Rice

Standard analysis
A multivariate approach
Interpretation of results

Balanced longitudinal data - Oranges

Initial analyses
Random coefficients and cubic smoothing splines

Multivariate animal genetics data - Sheep

Half-sib analysis
Animal model

15.1 Introduction

In this chapter we present the analysis of a variety of examples. The primary aim is to illustrate the capabilities of ASReml in the context of analysing real data sets. We also discuss the output produced by ASReml and indicate when problems may occur. Statistical concepts and issues are discussed as necessary but we stress that the analyses are illustrative, not prescriptive.

15.2 Split plot design - Oats

The first example involves the analysis of a split plot design originally presented by Yates (1935). The experiment was conducted to assess the effects on yield of three oat varieties (Golden Rain, Marvellous and Victory) with four levels of nitrogen application (0, 0.2, 0.4 and 0.6 cwt/acre). The field layout consisted of six blocks (labelled I, II, III, IV, V and VI) with three whole-plots per block each split into four sub-plots. The three varieties were randomly allocated to the three whole-plots while the four levels of nitrogen application were randomly assigned to the four sub-plots within each whole-plot. The data is presented in Table 15.1.

Table 15.1 A split-plot field trial of oat varieties and nitrogen application

block	variety	nitrogen			
		0.0cwt	0.2cwt	0.4cwt	0.6cwt
I	GR	111	130	157	174
	M	117	114	161	141
	V	105	140	118	156
II	GR	61	91	97	100
	M	70	108	126	149
	V	96	124	121	144
III	GR	68	64	112	86
	M	60	102	89	96
	V	89	129	132	124
IV	GR	74	89	81	122
	M	64	103	132	133
	V	70	89	104	117
V	GR	62	90	100	116
	M	80	82	94	126
	V	63	70	109	99
VI	GR	53	74	118	113
	M	89	82	86	104
	V	97	99	119	121

A standard analysis of these data recognises the two basic elements inherent in the experiment. These two aspects are firstly the stratification of the experiment units, that is the blocks, whole-plots and sub-plots, and secondly, the treatment

structure that is superimposed on the experimental material. The latter is of prime interest, in the presence of stratification. Thus the aim of the analysis is to examine the importance of the treatment effects while accounting for the stratification and restricted randomisation of the treatments to the experimental units. The ASReml input file is presented below.

```
split plot example
blocks 6 # Coded 1..6 in first data field of oats.asd
nitrogen !A 4 # Coded alphabetically
subplots * # Coded 1..4
variety !A 3 # Coded alphabetically
wplots * # Coded 1..3
yield
oats.asd !SKIP 2

yield ~ mu variety nitrogen variety.nitrogen !r blocks blocks.wplots
predict nitrogen # Print table of predicted nitrogen means
predict variety
predict variety nitrogen !SED
```

The data fields were `blocks`, `wplots`, `subplots`, `variety`, `nitrogen` and `yield`. The first five variables are factors that describe the stratification or experiment design and treatments. The standard split plot analysis is achieved by fitting the model terms `blocks` and `blocks.wplots` as random effects. The `blocks.wplots.subplots` term is not listed in the model because this interaction corresponds to the experimental units and is automatically included as the residual term. The fixed effects include the main effects of both `variety` and `nitrogen` and their interaction. The tables of predicted means and associated standard errors of differences (SEDs) have been requested. These are reported in the `.pvs` file. Abbreviated output is shown below.

Source	Model	terms	Gamma	Component	Comp/SE	% C
blocks	6	6	1.21116	214.477	1.27	0 P
blocks.wplots	18	18	0.598937	106.062	1.56	0 P
Variance	72	60	1.00000	177.083	4.74	0 P

Analysis of Variance	DF	F-incr
4 variety	2	1.49
2 nitrogen	3	37.69
8 variety.nitrogen	6	0.30

For simple variance component models such as the above, the default parameterisation for the variance component parameters is as the ratio to the residual

variance. Thus ASReml prints the variance component ratio and variance component for each term in the random model in the columns labelled `Gamma` and `Component` respectively.

The analysis of variance (ANOVA) is printed below this summary. The usual decomposition has three strata, with treatment effects separating into different strata as a consequence of the balanced design and the allocation of `variety` to whole-plots. In this balanced case, it is straightforward to derive the ANOVA estimates of the stratum variances from the above REML estimates of the variance components. That is

$$\begin{aligned} \text{blocks} &= 12\tilde{\sigma}_b^2 + 4\tilde{\sigma}_w^2 + \tilde{\sigma}^2 = 3175.1 \\ \text{blocks.wplots} &= 4\tilde{\sigma}_w^2 + \tilde{\sigma}^2 = 601.3 \\ \text{residual} &= \tilde{\sigma}^2 = 177.1 \end{aligned}$$

The default output for testing fixed effects used by ASReml is a table of so-called incremental F-tests. These F-tests are described in Section 6.10. The statistics are simply the appropriate Wald test statistics divided by the number of estimable effects for that term. In this example there are three terms included in the summary. The overall mean (denoted by `mu`) is of no interest for these data. The tests are sequential, that is the effect of each term is assessed by the change in sums of squares achieved by adding the term to the current model, defined by the model which includes those terms appearing above the current term given the variance parameters. For example, the test of `nitrogen` is calculated from the change in sums of squares for the two models `mu variety nitrogen` and `mu variety`. No refitting occurs, that is the variance parameters are held constant at the REML estimates obtained from the currently specified fixed model.

The incremental Wald statistics have an asymptotic χ^2 distribution, with degrees of freedom (df) given by the number of estimable effects (the number in the `DF` column). In this example, the incremental F-tests are numerically the same as the ANOVA F-tests, however ASReml currently has no facility for automatically determining the appropriate denominator df for testing fixed effects. This is a simple problem for balanced designs, such as the split plot design, but it is not straightforward to determine the relevant denominator df in unbalanced designs, such as the rat data set described in the next section.

Tables of predicted means are presented for the nitrogen, variety, and variety by nitrogen tables in the `.pvs` file. The qualifier `!SED` has been used on the third predict statement and so the matrix of SEDs for the variety by nitrogen table

is printed. For the first two predictions, the average SED is calculated from the average variance of differences. Note also that the order of the predictions (e.g. 0.6_cwt, 0.4_cwt 0.2_cwt 0_cwt for nitrogen) is simply the order those treatment labels were discovered in the data file.

```
Split plot analysis - oat Variety.Nitrogen          26 Aug 2002 00:26:33
                                                    oats
```

Ecode is E for Estimable, * for Not Estimable

```
----- 1 -----
variety          is averaged over fixed levels
Predicted values of yield
blocks          is ignored in the prediction (except where specifically included
wplots         is ignored in the prediction (except where specifically included

nitrogen        Predicted_Value Standard_Error Ecode
0.6_cwt         123.3889          7.1747 E
0.4_cwt         114.2222          7.1747 E
0.2_cwt         98.8889          7.1747 E
0_cwt           79.3889          7.1747 E
SED: Overall Standard Error of Difference  4.436
```

```
----- 2 -----
nitrogen        is averaged over fixed levels
Predicted values of yield
blocks          is ignored in the prediction (except where specifically included
wplots         is ignored in the prediction (except where specifically included

variety        Predicted_Value Standard_Error Ecode
Marvellous     109.7917          7.7975 E
Victory        97.6250          7.7975 E
Golden_rain    104.5000          7.7975 E
SED: Overall Standard Error of Difference  7.079
```

```
----- 3 -----
Predicted values of yield
blocks          is ignored in the prediction (except where specifically included
wplots         is ignored in the prediction (except where specifically included

variety        nitrogen        Predicted_Value Standard_Error Ecode
Marvellous     0.6_cwt          126.8333          9.1070 E
Victory        0.6_cwt          118.5000          9.1070 E
Golden_rain    0.6_cwt          124.8333          9.1070 E
Marvellous     0.4_cwt          117.1667          9.1070 E
Victory        0.4_cwt          110.8333          9.1070 E
Golden_rain    0.4_cwt          114.6667          9.1070 E
Marvellous     0.2_cwt          108.5000          9.1070 E
Victory        0.2_cwt           89.6667          9.1070 E
```

Golden_rain	0.2_cwt		98.5000	9.1070 E	
Marvellous	0_cwt		86.6667	9.1070 E	
Victory	0_cwt		71.5000	9.1070 E	
Golden_rain	0_cwt		80.0000	9.1070 E	
Predicted values with SED(PV)					
126.833					
118.500	9.71503				
124.833	9.71503	9.71503			
117.167	7.68295	9.71503	9.71503		
110.833	9.71503	7.68295	9.71503	9.71503	
114.667	9.71503	9.71503	7.68295	9.71503	
9.71503					
108.500	7.68295	9.71503	9.71503	7.68295	
9.71503	9.71503				
89.6667	9.71503	7.68295	9.71503	9.71503	
7.68295	9.71503	9.71503			
98.5000	9.71503	9.71503	7.68295	9.71503	
9.71503	7.68295	9.71503	9.71503		
86.6667	7.68295	9.71503	9.71503	7.68295	
9.71503	9.71503	7.68295	9.71503	9.71503	
71.5000	9.71503	7.68295	9.71503	9.71503	
7.68295	9.71503	9.71503	7.68295	9.71503	
9.71503					
80.0000	9.71503	9.71503	7.68295	9.71503	
9.71503	7.68295	9.71503	9.71503	7.68295	
9.71503	9.71503				
SED: Min	7.6830	Mean	9.1608	Max	9.7150

15.3 Unbalanced nested design - Rats

The second example we consider is a data set which illustrates some further aspects of testing fixed effects in linear mixed models. This example differs from the split plot example, as it is unbalanced and so more care is required in assessing the significance of fixed effects.

The experiment was reported by Dempster *et al.* (1984) and was designed to compare the effect of three doses of an experimental compound (control, low and high) on the maternal performance of rats. Thirty female rats (**dams**) were randomly split into three groups of 10 and each group randomly assigned to the three different doses. All pups in each litter were weighed. The litters differed in total size and in the numbers of males and females. Thus the additional covariate, **littersize** was included in the analysis. The differential effect of the compound on male and female pups was also of interest. Three litters had to be dropped from experiment, which meant that one dose had only 7 dams. The analysis

must account for the presence of between dam variation, but must also recognise the stratification of the experimental units (pups within litters) and that doses and littersize belong to the dam stratum. Table 15.2 presents an indicative AOV decomposition for this experiment.

Table 15.2 Rat data: AOV decomposition

stratum	decomposition	type	df or ne
constant	1	F	1
dams	dose	F	2
	littersize	F	1
	dam	R	27
dams.pups	sex	F	1
	dose.sex	F	2
error		R	

The dose and littersize effects are tested against the residual dam variation, while the remaining effects are tested against the residual within litter variation. The ASReml input to achieve this analysis is presented below.

```
Rats example
dose 3 !A
sex 2 !A
littersize
dam 27
pup 18
weight
rats.asd !DOPART 1 # Change DOPART argument to select each PART
!PART 1
weight ~ mu littersize dose sex dose.sex !r dam
!PART 2
weight ~ mu littersize sex dose !r dam
!PART 3
weight ~ mu sex dose littersize !r dam
!PART 4
weight ~ mu littersize dose sex !r dam
```

The input file contains an example of the use of the `!DOPART` qualifier. Its argument specifies which part to execute. We will discuss the models in the four parts. The full model is fitted in part 1. Abbreviated output from this analysis is presented below.

```

LogL= 74.2174      S2= 0.19670          315 df   0.1000      1.000
LogL= 79.1577      S2= 0.18751          315 df   0.1488      1.000
LogL= 83.9407      S2= 0.17755          315 df   0.2446      1.000
LogL= 86.8093      S2= 0.16903          315 df   0.4254      1.000
LogL= 87.2249      S2= 0.16594          315 df   0.5521      1.000
LogL= 87.2398      S2= 0.16532          315 df   0.5854      1.000
LogL= 87.2398      S2= 0.16530          315 df   0.5867      1.000
Final parameter values                                0.58667      1.0000

Source          Model terms      Gamma      Component      Comp/SE      % C
dam              27      27  0.586674      0.969769E-01      2.92      0 P
Variance         322     315  1.00000      0.165299          12.09      0 P

Analysis of Variance      DF      F-incr
  3 littersize            1      27.99
  1 dose                   2      12.15
  2 sex                     1      57.96
  8 dose.sex               2       0.40

```

The iterative sequence has converged and the variance component parameter for `dam` hasn't changed for the last two iterations. The incremental Wald tests indicate that the interaction between `dose` and `sex` is not significant. Since these tests are sequential, the test for the `dose.sex` term is appropriate as it respects marginality with both the main effects of dose and sex fitted before the interaction.

Before proceeding we note the `.res` file reports several possible outliers, in particular unit 66 has a standardised residual of -8.80 (see Figure 15.1). The weight of this female rat, within litter 9 is only 3.68, compared to weights of 7.26 and 6.58 for two other female sibling pups. This weight appears erroneous, but without knowledge of the actual experiment we retain the observation in the following.

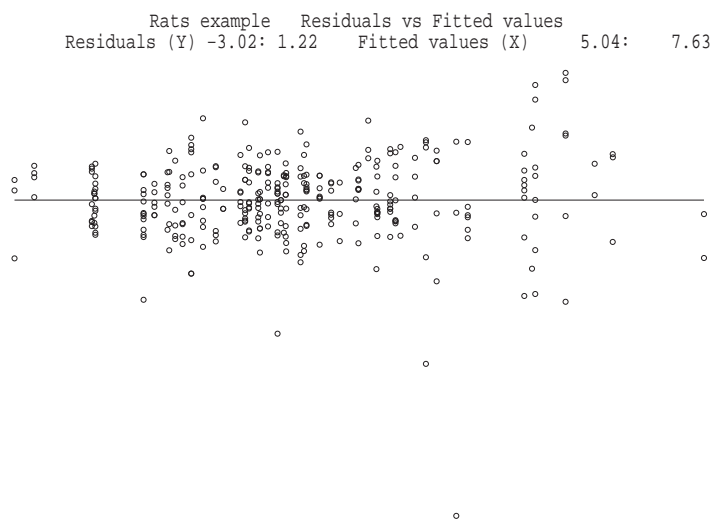


Figure 15.1 Residual plot for the rat data

We refit the model without the `dose.sex` term. Note that the variance parameters are re-estimated, though there is little change from the previous analysis. The output below presents the incremental Wald tests for this analysis.

Source	Model	terms	Gamma	Component	Comp/SE	% C
dam	27	27	0.595157	0.979179E-01	2.93	0 P
Variance	322	317	1.00000	0.164524	12.13	0 P

Analysis of Variance	DF	F-incr
7 mu	1	8981.48
3 littersize	1	27.85
2 sex	1	59.54
1 dose	2	11.42

The three remaining fixed effects appear significant, though an additional two models (part 3 and part 4) must be fitted to properly test the fixed effects in the model. Table 15.3 presents the summary of these analyses.

The second column of F-tests pertains to a model which (wrongly) excludes the `dam` term. The impact of deleting the `dam` term on the F-tests for the fixed effects is substantial and not surprising. This example reinforces the importance of preserving the strata of the design when assessing the significance of fixed effects.

Table 15.3: Summary of F-tests of fixed effects adjusted for the other factors for the rat data

term	df	F-test	
		with dam	without dam
littersize	1	46.43	146.50
dose	1	11.42	58.43
sex	1	58.27	24.52

15.4 Source of variability in unbalanced data - Volts

In this example we illustrate an analysis of unbalanced data in which the main aim is to determine the sources of variation rather than assess the significance of imposed treatments. The data are taken from Cox and Snell (1981) and involve an experiment to examine the variability in the production of car voltage regulators. Standard production of regulators involves two steps. Regulators are taken from the production line and passed to a setting station which adjusts the regulator to operate within a specified range of voltage. From the setting station the regulator is then passed to a testing station where it is tested and returned if outside the required range.

A total of 64 regulators were tested at four testing stations (`teststat`). The voltage for individual regulators was set at a total of 10 setting stations (`setstat`). A variable number of regulators (between 4 to 8) were set at each station, however each regulator was tested at every testing station. The ASReml input file is presented below

```
Voltage data
teststat 4 # 4 testing stations tested each regulator
setstat !A # 10 setting stations each set 4-8 regulators
regulatr 8 # regulators numbered within setting stations
voltage
5-3-6.asd !skip 1
voltage ~ mu !r setstat setstat.regulatr teststat setstat.teststat
0 0 0
```

The factor `regulatr` numbers the regulators within each setting station. Thus the term `setstat.regulatr` allows for differential effects of each regulator, while

the other terms examine the effects of the setting and testing stations and possible interaction. The abbreviated output is given below

```

LogL= 188.604      S2= 0.67074E-01      255 df
LogL= 199.530      S2= 0.59303E-01      255 df
LogL= 203.007      S2= 0.52814E-01      255 df
LogL= 203.240      S2= 0.51278E-01      255 df
LogL= 203.242      S2= 0.51141E-01      255 df
LogL= 203.242      S2= 0.51140E-01      255 df

Source           Model  terms      Gamma      Component    Comp/SE    % C
setstat          10    10  0.233418    0.119371E-01  1.35    0 P
setstat.regulatr 80    64  0.601817    0.307771E-01  3.64    0 P
teststat         4     4  0.642752E-01 0.328706E-02  0.98    0 P
setstat.teststat 40    40  0.100000E-08 0.511404E-10  0.00    0 B
Variance         256   255  1.00000     0.511404E-01  9.72    0 P
Warning: Code B - fixed at a boundary (!GP)      F - fixed by user
           ? - liable to change from P to B      P - positive definite
           C - Constrained by user (!VCC)        U - unbounded
           S - Singular Information matrix

```

The convergence criteria has been satisfied after six iterations. A warning message is printed below the summary of the variance components because the variance component for the `setstat.teststat` term has been fixed near the boundary. The default constraint for variance components (`!GP`) is to ensure that the REML estimate remains positive. Under this constraint, if an update for any variance component results in a negative value then `ASReml` sets that variance component to a small positive value. If this occurs in subsequent iterations the parameter is fixed to a small positive value and the code B replaces P in the C column of the summary table. The default constraint can be overridden using the `!GU` qualifier, but it is not generally recommended for standard analyses.

Figure 15.2 presents the residual plot which indicates two unusual data values. These values are successive observations, namely observation 210 and 211, being testing stations 2 and 3 for setting station 9(J), regulator 2. These observations will not be dropped from the following analyses for consistency with other analyses conducted by Cox and Snell (1981) and in the `GENSTAT` manual.

The REML log-likelihood from the model without the `setstat.teststat` term was 203.242, the same as the REML log-likelihood for the previous model. Table 15.4 presents a summary of the REML log-likelihood ratio for the remaining terms in the model. The summary of the `ASReml` output for the current model is given below. The column labelled `Comp/SE` is printed by `ASReml` to give a guide

```
ltage example 5-3-6 from the GENSTAT REML manual  Residuals vs Fitted valu
Residuals (Y) -1.08: 1.45  Fitted values (X)  15.56: 16.81
```

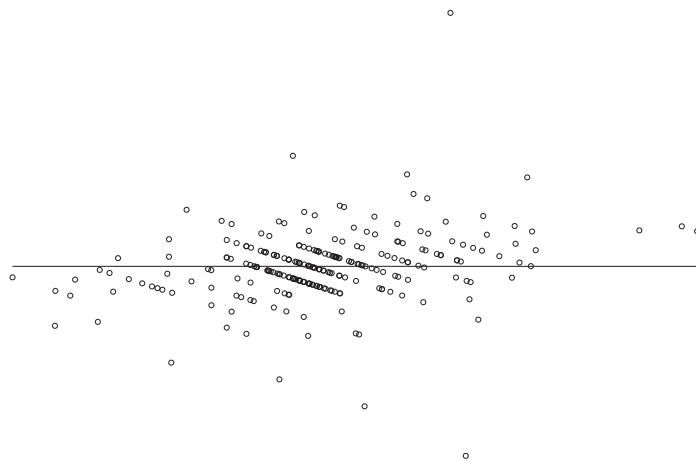


Figure 15.2 Residual plot for the voltage data

as to the significance of the variance component for each term in the model. The statistic is simply the REML estimate of the variance component divided by the square root of the diagonal element (for each component) of the inverse of the average information matrix. The diagonal elements of the expected (not the average) information matrix are the asymptotic variances of the REML estimates of the variance parameters. These **Comp/SE** statistics cannot be used to test the null hypothesis that the variance component is zero. If we had used this crude measure then the conclusions would have been inconsistent with the conclusions obtained from the REML log-likelihood ratio (see Table 15.4).

Source	Model	terms	Gamma	Component	Comp/SE	% C
setstat	10	10	0.233417	0.119370E-01	1.35	0 P
setstat.regulatr	80	64	0.601817	0.307771E-01	3.64	0 P
teststat	4	4	0.642752E-01	0.328705E-02	0.98	0 P
Variance	256	255	1.00000	0.511402E-01	9.72	0 P

15.5 Balanced repeated measures - Height

The data for this example is taken from the GENSTAT manual. It consists of a total of 5 measurements of height (cm) taken on 14 plants. The 14 plants were either diseased or healthy and were arranged in a glasshouse in a completely

Table 15.4: REML log-likelihood ratio for the variance components in the voltage data

terms	REML log-likelihood	$-2\times$ difference	P-value
– setstat	200.31	5.864	.0077
– setstat.regulatr	184.15	38.19	.0000
– teststat	199.71	7.064	.0039

random design. The heights were measured 1, 3, 5, 7 and 10 weeks after the plants were placed in the glasshouse. There were 7 plants in each treatment. The data are depicted in Figure 15.3 obtained by qualifier line

```
!Y y1 !G tmt !JOIN
```

in the following multivariate ASReml job.

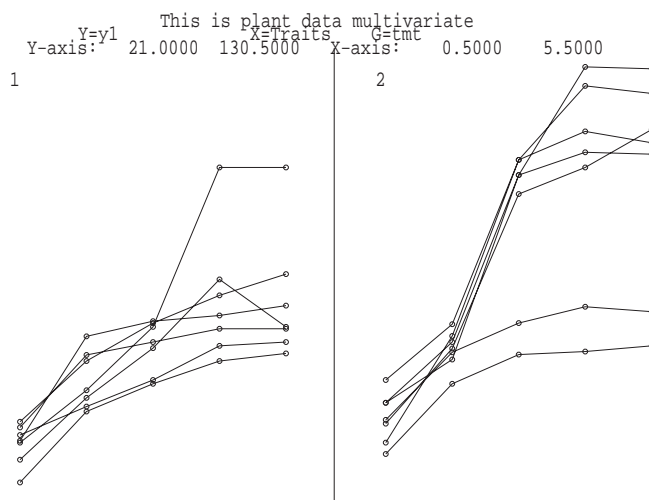


Figure 15.3 Trellis plot of the height for each of 14 plants

In the following we illustrate how various repeated measures analyses can be conducted in ASReml. For these analyses it is convenient to arrange the data in a multivariate form, with 7 fields representing the plant number, treatment identification and the 5 heights. The ASReml input file, up to the specification of the R structure is given by

```

This is plant data multivariate
tmt !A # Diseased Healthy
plant 14
y1 y3 y5 y7 y10
plantmult.asd !skip 1 !ASUV

```

The focus is modelling of the error variance for the data. Specifically we fit the multivariate regression model given by

$$\mathbf{Y} = \mathbf{DT} + \mathbf{E} \quad (15.1)$$

where $\mathbf{Y}^{14 \times 5}$ is the matrix of heights, $\mathbf{D}^{14 \times 2}$ is the design matrix, $\mathbf{T}^{2 \times 5}$ is the matrix of fixed effects and $\mathbf{E}^{14 \times 5}$ is the matrix of errors. The heights taken on the same plants will be correlated and so we assume that

$$\text{var}(\text{vec}(\mathbf{E})) = \mathbf{I}_{14} \otimes \mathbf{\Sigma} \quad (15.2)$$

where $\mathbf{\Sigma}^{5 \times 5}$ is a symmetric positive definite matrix.

The variance models used for $\mathbf{\Sigma}$ are given in Table 15.5. These represent some commonly used models for the analysis of repeated measures data (see Wolfinger, 1986). The variance models are fitted by changing the last four lines of the input file. The sequence of commands for the first model fitted is given below

```

y1 y3 y5 y7 y10 ~ Trait tmt Tr.tmt !r units
1 2 0
14
Tr

```

Table 15.5 Summary of variance models fitted to the plant data

model	number of parameters	REML log-likelihood	BIC
Uniform	2	-196.88	401.95
Power	2	-182.98	374.15
Heterogeneous Power	6	-171.50	367.57
Antedependence (order 1)	9	-160.37	357.51
Unstructured	15	-158.04	377.50

The split plot in time model can be fitted in two ways, either by fitting a `units` term plus an independent residual as above, or by specifying a CORU variance model for the R -structure as follows

```
y1 y3 y5 y7 y10 ~ Trait tmt Tr.tmt
1 2 0
14
Tr 0 CORU .5
```

The two forms for Σ are given by

$$\begin{aligned}\Sigma &= \sigma_1^2 \mathbf{J} + \sigma_2^2 \mathbf{I}, & \text{units} \\ \Sigma &= \sigma_e^2 \mathbf{I} + \sigma_e^2 \rho (\mathbf{J} - \mathbf{I}), & \text{CORU}\end{aligned}\quad (15.3)$$

It follows that

$$\begin{aligned}\sigma_e^2 &= \sigma_1^2 + \sigma_2^2 \\ \rho &= \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\end{aligned}\quad (15.4)$$

Portions of the two outputs are given below. The REML log-likelihoods for the two models are the same and it is easy to verify that the REML estimates of the variance parameters satisfy (15.4), viz. $\sigma_e^2 = 286.310 \approx 159.858 + 126.528 = 286.386$; $159.858/286.386 = 0.558191$.

```
#
#
# !r units
#
LogL=-204.593      S2=  224.61          60 df   0.1000      1.000
LogL=-201.233      S2=  186.52          60 df   0.2339      1.000
LogL=-198.453      S2=  155.09          60 df   0.4870      1.000
LogL=-197.041      S2=  133.85          60 df   0.9339      1.000
LogL=-196.881      S2=  127.56          60 df   1.204       1.000
LogL=-196.877      S2=  126.53          60 df   1.261       1.000
Final parameter values                1.2634      1.0000

Source           Model  terms   Gamma      Component  Comp/SE  % C
units            14    14     1.26342    159.858    2.11    0 P
Variance         70    60     1.00000    126.528    4.90    0 P
#
# CORU
#
LogL=-196.975      S2=  264.10          60 df   1.000       0.5000
LogL=-196.924      S2=  270.14          60 df   1.000       0.5178
LogL=-196.886      S2=  278.58          60 df   1.000       0.5400
```

```

LogL=-196.877      S2= 286.23          60 df    1.000    0.5580
LogL=-196.877      S2= 286.31          60 df    1.000    0.5582
Final parameter values                                1.0000    0.55819

```

Source	Model	terms	Gamma	Component	Comp/SE	% C
Variance	70	60	1.00000	286.310	3.65	0 P
Residual	CORRelat	5	0.558191	0.558191	4.28	0 U

A more realistic model for repeated measures data would allow the correlations to decrease as the lag increases such as occurs with the first order autoregressive model. However, since the heights are not measured at equally spaced time points we use the EXP model. The correlation function is given by

$$\rho(u) = \phi^u$$

where u is the time lag in weeks. The coding for this is

```

y1 y3 y5 y7 y10 ~ Trait tmt Tr.tmt
1 2 0          # One error structure in two dimensions
14            # Outer dimension: 14 plants
Tr 0 EXP .5
1 3 5 7 10    # Time coordinates

```

A portion of the output is given below

```

LogL=-183.734      S2= 435.58          60 df    1.000    0.9500
LogL=-183.255      S2= 370.40          60 df    1.000    0.9388
LogL=-183.010      S2= 321.50          60 df    1.000    0.9260
LogL=-182.980      S2= 298.84          60 df    1.000    0.9179
LogL=-182.979      S2= 302.02          60 df    1.000    0.9192
Final parameter values                                1.0000    0.91897

```

Source	Model	terms	Gamma	Component	Comp/SE	% C
Variance	70	60	1.00000	302.021	3.11	0 P
Residual	POW-EXP	5	0.918971	0.918971	29.53	0 U

When fitting power models be careful to ensure the scale of the defining variate, here `time`, does not result in an estimate of ϕ too close to 1. For example, use of days in this example would result in an estimate for ϕ of about .993.

The residual plot from this analysis is presented in Figure 15.4. This suggests increasing variance over time. This can be modelled by using the EXPH model,

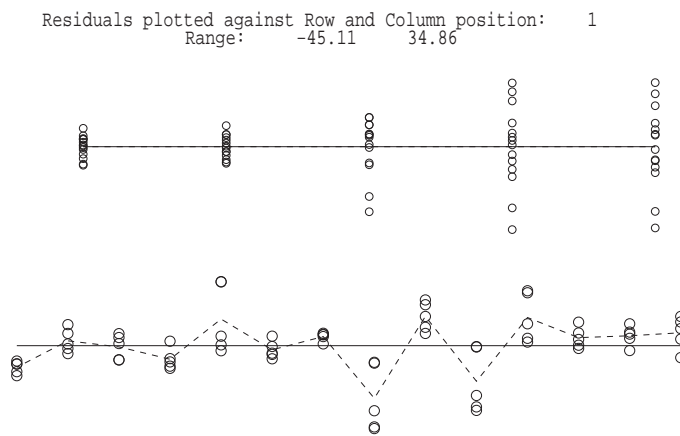


Figure 15.4 Residual plots for the EXP variance model for the plant data

which models Σ by

$$\Sigma = D^{0.5} C D^{0.5}$$

where D is a diagonal matrix of variances and C is a correlation matrix with elements given by $c_{ij} = \phi^{|t_i - t_j|}$. The coding for this is

```
y1 y3 y5 y7 y10 ~ Trait tmt Tr.tmt
1 2 0
14 !S2==1
Tr 0 EXPH .5 100 200 300 300 300
1 3 5 7 10
```

Note that it is necessary to fix the scale parameter to 1 (!S2==1) to ensure that the elements of D are identifiable. Abbreviated output from this analysis is given below.

```
1 LogL=-195.598 S2= 1.0000 60 df : 1 components constrained
2 LogL=-179.036 S2= 1.0000 60 df
3 LogL=-175.483 S2= 1.0000 60 df
4 LogL=-173.128 S2= 1.0000 60 df
5 LogL=-171.980 S2= 1.0000 60 df
6 LogL=-171.615 S2= 1.0000 60 df
7 LogL=-171.527 S2= 1.0000 60 df
```

```

      8 LogL=-171.504      S2=  1.0000      60 df
      9 LogL=-171.498      S2=  1.0000      60 df
     10 LogL=-171.496      S2=  1.0000      60 df

```

Source	Model	terms	Gamma	Component	Comp/SE	% C
Residual	POW-EXP	5	0.906917	0.906917	21.89	0 U
Residual	POW-EXP	5	60.9599	60.9599	2.12	0 U
Residual	POW-EXP	5	72.9904	72.9904	1.99	0 U
Residual	POW-EXP	5	309.259	309.259	2.22	0 U
Residual	POW-EXP	5	436.380	436.380	2.52	0 U
Residual	POW-EXP	5	382.369	382.369	2.74	0 U

Covariance/Variance/Correlation Matrix POWER

61.11	0.8227	0.6769	0.5569	0.4156
54.88	72.80	0.8227	0.6769	0.5051
93.12	123.5	309.7	0.8227	0.6140
91.02	120.7	302.7	437.1	0.7462
63.57	84.34	211.4	305.3	382.9

Analysis of Variance	DF	F-incr
8 Trait	5	127.95
1 tmt	1	0.00
9 Tr.tmt	4	4.75

The last two models we fit are the antedependence model of order 1 and the unstructured model. These require, as starting values the lower triangle of the full variance matrix. We use the REML estimate of Σ from the heterogeneous power model shown in the previous output. The antedependence model models Σ by the inverse cholesky decomposition

$$\Sigma^{-1} = UDU'$$

where D is a diagonal matrix and U is a unit upper triangular matrix. For an antedependence model of order q , then $u_{ij} = 0$ for $j > i + q - 1$. The antedependence model of order 1 has 9 parameters for these data, 5 in D and 4 in U . The input is given by

```

y1 y3 y5 y7 y10 ~ Trait tmt Tr.tmt
1 2 0
14 !S2==1
Tr 0 ANTE
60.16
54.65      73.65
91.50      123.3      306.4
89.17      120.2      298.6      431.8
62.21      83.85      208.3      301.2      379.8

```

The abbreviated output file is given below

```

1 LogL=-171.501      S2=  1.0000      60 df
2 LogL=-170.097      S2=  1.0000      60 df
3 LogL=-166.085      S2=  1.0000      60 df
4 LogL=-161.335      S2=  1.0000      60 df
5 LogL=-160.407      S2=  1.0000      60 df
6 LogL=-160.370      S2=  1.0000      60 df
7 LogL=-160.369      S2=  1.0000      60 df

Source      Model terms      Gamma      Component      Comp/SE      % C
Residual ANTE=UDU      1  0.268657E-01  0.268657E-01  2.44  0 U
Residual ANTE=UDU      1 -0.628413      -0.628413      -2.55  0 U
Residual ANTE=UDU      2  0.372801E-01  0.372801E-01  2.41  0 U
Residual ANTE=UDU      2 -1.49108      -1.49108      -2.54  0 U
Residual ANTE=UDU      3  0.599632E-02  0.599632E-02  2.43  0 U
Residual ANTE=UDU      3 -1.28041      -1.28041      -6.19  0 U
Residual ANTE=UDU      4  0.789713E-02  0.789713E-02  2.44  0 U
Residual ANTE=UDU      4 -0.967815      -0.967815      -15.40  0 U
Residual ANTE=UDU      5  0.390635E-01  0.390635E-01  2.45  0 U
Covariance/Variance/Correlation Matrix ANTE=UDU'
37.20      0.5946      0.3549      0.3114      0.3040
23.38      41.55      0.5968      0.5237      0.5112
34.83      61.89      258.9      0.8775      0.8565
44.58      79.22      331.4      550.8      0.9761
43.14      76.67      320.7      533.0      541.4

Analysis of Variance      DF      F-incr
8 Trait      5      188.84
1 tmt      1      4.14
9 Tr.tmt      4      3.91

```

The iterative sequence converged and the antedependence parameter estimates are printed successively for each time, the element of \mathbf{D} and then the row of \mathbf{U} . I.e.

$$\mathbf{D} = \text{diag} \begin{bmatrix} 0.0269 \\ 0.0373 \\ 0.0060 \\ 0.0079 \\ 0.0391 \end{bmatrix}, \mathbf{U} = \begin{bmatrix} 1 & -0.6284 & 0 & 0 & 0 \\ 0 & 1 & -1.4911 & 0 & 0 \\ 0 & 0 & 1 & -1.2804 & 0 \\ 0 & 0 & 0 & 1 & -0.9678 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Finally the input and output files for the unstructured model are presented below. The REML estimate of Σ from the ANTE model is used to provide starting values.

```

#####
#input
#####
y1 y3 y5 y7 y10 ~ Trait tmt Tr.tmt
1 2 0
14 !S2==1
Tr 0 US
  37.20
  23.38      41.55
  34.83      61.89      258.9
  44.58      79.22      331.4      550.8
  43.14      76.67      320.7      533.0      541.4
#####
# output
#####
1 LogL=-160.368      S2=  1.0000      60 df
2 LogL=-159.027      S2=  1.0000      60 df
3 LogL=-158.247      S2=  1.0000      60 df
4 LogL=-158.040      S2=  1.0000      60 df
5 LogL=-158.036      S2=  1.0000      60 df

Source      Model  terms      Gamma      Component      Comp/SE      % C
Residual US=UnStr      1      37.2262      37.2262      2.45      0 U
Residual US=UnStr      1      23.3935      23.3935      1.77      0 U
Residual US=UnStr      2      41.5195      41.5195      2.45      0 U
Residual US=UnStr      1      51.6524      51.6524      1.61      0 U
Residual US=UnStr      2      61.9169      61.9169      1.78      0 U
Residual US=UnStr      3      259.121      259.121      2.45      0 U
Residual US=UnStr      1      70.8113      70.8113      1.54      0 U
Residual US=UnStr      2      57.6146      57.6146      1.23      0 U
Residual US=UnStr      3      331.807      331.807      2.29      0 U
Residual US=UnStr      4      551.507      551.507      2.45      0 U
Residual US=UnStr      1      73.7857      73.7857      1.60      0 U
Residual US=UnStr      2      62.5691      62.5691      1.33      0 U
Residual US=UnStr      3      330.851      330.851      2.29      0 U
Residual US=UnStr      4      533.756      533.756      2.42      0 U
Residual US=UnStr      5      542.175      542.175      2.45      0 U
Covariance/Variance/Correlation Matrix US=UnStructu
  37.23      0.5950      0.5259      0.4942      0.5194
  23.39      41.52      0.5969      0.3807      0.4170
  51.65      61.92      259.1      0.8777      0.8827
  70.81      57.61      331.8      551.5      0.9761
  73.79      62.57      330.9      533.8      542.2

```

The antedependence model of order 1 is clearly more parsimonious than the unstructured model. Table 15.6 presents the incremental Wald tests for each of the variance models. There is a surprising level of discrepancy between models for the Wald tests. The main effect of treatment is significant for the uniform, power and antedependence models.

Table 15.6: Summary of Wald test for fixed effects for variance models fitted to the plant data

model	treatment (df=1)	treatment.time (df=4)
Uniform	9.41	5.10
Power	6.86	6.13
Heterogeneous power	0.00	4.81
Antedependence (order 1)	4.14	3.96
Unstructured	1.71	4.46

15.6 Spatial analysis of a field experiment - Barley

In this section we illustrate the ASReml syntax for performing spatial and incomplete block analysis of a field experiment. There has been a large amount of interest in developing techniques for the analysis of spatial data both in the context of field experiments and geostatistical data (see for example, Cullis and Gleeson, 1991; Cressie, 1991; Gilmour *et al.*, 1997). This example illustrates the analysis of 'so-called' regular spatial data, in which the data is observed on a lattice or regular grid. This is typical of most small plot designed field experiments. Spatial data is often irregularly spaced, either by design or because of the observational nature of the study. The techniques we present in the following can be extended for the analysis of irregularly spaced spatial data, though, larger spatial data sets may be computationally challenging, depending on the degree of irregularity or models fitted.

The data we consider is taken from Gilmour *et al.* (1995) and involves a field experiment designed to compare the performance of 25 varieties of barley. The experiment was conducted at Slate Hall Farm, UK in 1976, and was designed as a balanced lattice square with replicates laid out as shown in Table 15.7. The data fields were `Rep`, `RowBlk`, `ColBlk`, `row`, `column` and `yield`. Lattice row and column numbering is typically within replicates and so the terms specified in the linear model to account for the lattice row and lattice column effects would be `Rep.latticerow` `Rep.latticecolumn`. However, in this example lattice rows and columns are both numbered from 1 to 30 across replicates (see Table 15.7). The terms in the linear model are therefore simply `RowBlk` `ColBlk`. Additional fields `row` and `column` indicate the spatial layout of the plots.

The ASReml input file is presented below. Three models have been fitted to these data. The lattice analysis is included for comparison in PART 3. In PART 1 we use the separable first order autoregressive model to model the variance structure of the plot errors. Gilmour *et al.* (1997) suggest this is often a useful model to commence the spatial modelling process. The form of the variance matrix for the plot errors (R structure) is given by

$$\sigma^2 \Sigma = \sigma^2 (\Sigma_c \otimes \Sigma_r) \quad (15.5)$$

where Σ_c and Σ_r are 15×15 and 10×10 matrix functions of the column (ϕ_c) and row (ϕ_r) autoregressive parameters respectively.

Table 15.7 Field layout of Slate Hall Farm experiment

Column - Replicate levels															
Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
2	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
3	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
4	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
5	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
6	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
7	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
8	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
9	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
10	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
Column - Rowblk levels															
Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	11	11	11	11	11	21	21	21	21	21
2	2	2	2	2	2	12	12	12	12	12	22	22	22	22	22
3	3	3	3	3	3	13	13	13	13	13	23	23	23	23	23
4	4	4	4	4	4	14	14	14	14	14	24	24	24	24	24
5	5	5	5	5	5	15	15	15	15	15	25	25	25	25	25
6	6	6	6	6	6	16	16	16	16	16	26	26	26	26	26
7	7	7	7	7	7	17	17	17	17	17	27	27	27	27	27
8	8	8	8	8	8	18	18	18	18	18	28	28	28	28	28
9	9	9	9	9	9	19	19	19	19	19	29	29	29	29	29
10	10	10	10	10	10	20	20	20	20	20	30	30	30	30	30
Column - Colblk levels															
Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
7	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
8	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
9	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
10	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Gilmour *et al.* (1997) recommend revision of the current spatial model based on the use of diagnostics such as the sample variogram of the residuals (from the current model). This diagnostic and a summary of row and column residual trends are produced by default with graphical versions of ASReml when a spatial model has been fitted to the errors. It can be suppressed, by the use of the `-n` option on the command line. We have produced the following plots by use of the `-g22` option.

```
Slate Hall example
Rep 6      # Six replicates of 5x5 plots in 2x3 arrangement
RowBlk 30  # Rows within replicates numbered across replicates
ColBlk 30  # Columns within replicates numbered across replicates
row 10     # Field row
column 15  # Field column
variety 25
yield
shf.asd !skip 1 !DOPART 1
!PART 1 # AR1 x AR1
y ~ mu var
predict var
1 2
15 column AR1 0.1 # Second field is specified so ASReml can sort
10 row AR1 0.1    # records properly into field order

!PART 2 # AR1 x AR1 + units
y ~ mu var !r units
predict var
1 2
15 column AR1 0.1
10 row AR1 0.1

!PART 3 # incomplete blocks
y ~ mu var !r Rep Rowblk Colblk
predict var
```

Abbreviated ASReml output file is presented below. The iterative sequence has converged to column and row correlation parameters of (.68377,.45859) respectively. The plot size and orientation is not known and so it is not possible to ascertain whether these values are spatially sensible. It is generally found that the closer the plot centroids, the higher the spatial correlation. This is not always the case and if the highest between plot correlation relates to the larger spatial distance then this may suggest the presence of extraneous variation (see Gilmour *et al.*, 1997), for example. Figure 15.5 presents a plot of the sample variogram of the residuals from this model. The plot appears in reasonable agreement with the model.

The next model includes a measurement error or nugget effect component. That is the variance model for the plot errors is now given by

$$\sigma^2 \Sigma = \sigma^2 (\Sigma_c \otimes \Sigma_r) + \psi I_{150} \quad (15.6)$$

where ψ is the ratio of nugget variance to error variance (σ^2). The abbreviated output for this model is given below. There is a significant improvement in the REML log-likelihood with the inclusion of the nugget effect (see Table 15.8).

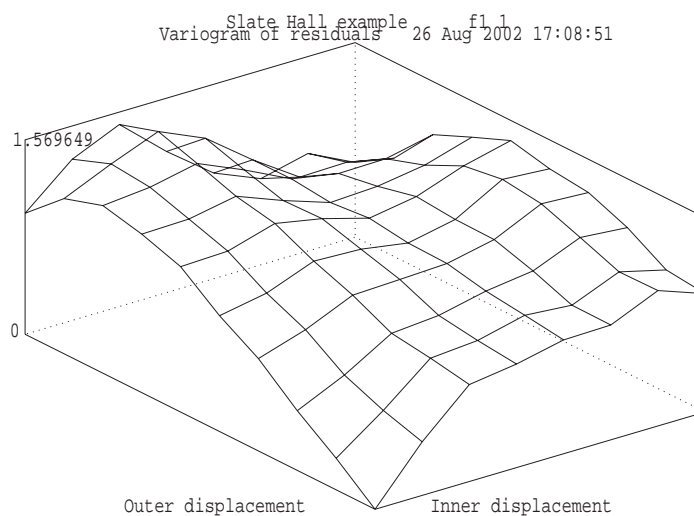


Figure 15.5 Sample variogram of the AR1 \times AR1 model for the Slate Hall data

```
# AR1 x AR1
#
 1 LogL=-739.681   S2= 36034.   125 df   1.000   0.1000   0.1000
 2 LogL=-714.340   S2= 28109.   125 df   1.000   0.4049   0.1870
 3 LogL=-703.338   S2= 29914.   125 df   1.000   0.5737   0.3122
 4 LogL=-700.371   S2= 37464.   125 df   1.000   0.6789   0.4320
 5 LogL=-700.324   S2= 38602.   125 df   1.000   0.6838   0.4542
 6 LogL=-700.322   S2= 38735.   125 df   1.000   0.6838   0.4579
 7 LogL=-700.322   S2= 38754.   125 df   1.000   0.6838   0.4585
 8 LogL=-700.322   S2= 38757.   125 df   1.000   0.6838   0.4586
Final parameter values           1.0000   0.68377   0.45861
```

Source	Model	terms	Gamma	Component	Comp/SE	% C
Variance	150	125	1.00000	38756.6	5.00	0 P
Residual	AR=AutoR	15	0.683767	0.683767	10.80	0 U
Residual	AR=AutoR	10	0.458607	0.458607	5.55	0 U

```

Analysis of Variance
  8 mu
  6 variety

# AR1 x AR1 + units
  1 LogL=-740.735 S2= 33225. 125 df : 2 components constrained
  2 LogL=-723.595 S2= 11661. 125 df : 1 components constrained
  3 LogL=-698.498 S2= 46239. 125 df
  4 LogL=-696.847 S2= 44725. 125 df
  5 LogL=-696.823 S2= 45563. 125 df
  6 LogL=-696.823 S2= 45753. 125 df
  7 LogL=-696.823 S2= 45796. 125 df

Source      Model terms      Gamma      Component      Comp/SE      % C
units       150 150 0.106154      4861.48      2.72 0 P
Variance    150 125 1.00000      45796.3      2.74 0 P
Residual    AR=AutoR 15 0.843795      0.843795      12.33 0 U
Residual    AR=AutoR 10 0.682686      0.682686      6.68 0 U

Analysis of Variance
  8 mu
  6 variety

```

The lattice analysis (with recovery of between block information) is presented below. This variance model is not competitive with the preceding spatial models. The models can be formally compared using the BIC values for example.

```

# IB analysis
  1 LogL=-734.184 S2= 26778. 125 df
  2 LogL=-720.060 S2= 16591. 125 df
  3 LogL=-711.119 S2= 11173. 125 df
  4 LogL=-707.937 S2= 8562.4 125 df
  5 LogL=-707.786 S2= 8091.2 125 df
  6 LogL=-707.786 S2= 8061.8 125 df
  7 LogL=-707.786 S2= 8061.8 125 df

Source      Model terms      Gamma      Component      Comp/SE      % C
Rep         6 6 0.528714      4262.39      0.62 0 P
RowBlk      30 30 1.93444      15595.1      3.06 0 P
ColBlk      30 30 1.83725      14811.6      3.04 0 P
Variance    150 125 1.00000      8061.81      6.01 0 P

Analysis of Variance
  8 mu
  6 variety

```

Finally, we present portions of the .pvs files to illustrate the prediction facility of ASReml. The first five and last three variety means are presented for illustration. The overall SED printed is the square root of the average variance of difference between the variety means. The two spatial analyses have a range of SEDs which are available if the !SED qualifier is used. All variety comparisons have the same SED from the third analysis as the design is a balanced lattice square. The F-test statistics for the spatial models are greater than for the lattice analysis. We note the F-test for the AR1×AR1 + units model is smaller than the F-test for the AR1×AR1.

Predicted values of yield

#AR1 x AR1

variety	Predicted_Value	Standard_Error	Ecode
1.0000	1257.9763	64.6146	E
2.0000	1501.4483	64.9783	E
3.0000	1404.9874	64.6260	E
4.0000	1412.5674	64.9027	E
5.0000	1514.4764	65.5889	E
.	.	.	.
23.0000	1311.4888	64.0767	E
24.0000	1586.7840	64.7043	E
25.0000	1592.0204	63.5939	E
SED: Overall Standard Error of Difference			59.05

#AR1 x AR1 + units

variety	Predicted_Value	Standard_Error	Ecode
1.0000	1245.5843	97.8591	E
2.0000	1516.2331	97.8473	E
3.0000	1403.9863	98.2398	E
4.0000	1404.9202	97.9875	E
5.0000	1471.6197	98.3607	E
.	.	.	.
23.0000	1316.8726	98.0402	E
24.0000	1557.5278	98.1272	E
25.0000	1573.8920	97.9803	E
SED: Overall Standard Error of Difference			60.51

IB

Rep	is ignored in the prediction
RowBlk	is ignored in the prediction
ColBlk	is ignored in the prediction

variety	Predicted_Value	Standard_Error	Ecode
1.0000	1283.5870	60.1994	E
2.0000	1549.0133	60.1994	E
3.0000	1420.9307	60.1994	E
4.0000	1451.8554	60.1994	E
5.0000	1533.2749	60.1994	E

```

      .
23.0000      1329.1088      60.1994 E
24.0000      1546.4699      60.1994 E
25.0000      1630.6285      60.1994 E
SED: Overall Standard Error of Difference      62.02

```

Notice the differences in SE and SED associated with the various models. Choosing a model on the basis of smallest SE or SED is not recommended because the model is not necessarily fitting the variability present in the data.

Table 15.8 Summary of models for the Slate Hall data

model	REML log-likelihood	number of parameters	F-test	SED
AR1×AR1	-700.32	3	13.04	59.0
AR1×AR1 + units	-696.82	4	10.22	60.5
IB	-707.79	4	8.84	62.0

15.7 Unreplicated early generation variety trial - Wheat

To further illustrate the approaches presented in the previous section, we consider an unreplicated field experiment conducted at Tullibigeal situated in southwestern NSW. The trial was an S1 (early stage) wheat variety evaluation trial and consisted of 525 test lines which were randomly assigned to plots in a 67 by 10 array. There was a check plot variety every 6 plots within each column. That is the check variety was sown on rows 1,7,13,...,67 of each column. This variety was numbered 526. A further 6 replicated commercially available varieties (numbered 527 to 532) were also randomly assigned to plots with between 3 to 5 plots of each. The aim of these trials is to identify and retain the top, say 20% of lines for further testing. Cullis *et al.* (1989) considered the analysis of early generation variety trials, and presented a one-dimensional spatial analysis which was an extension of the approach developed by Gleeson and Cullis (1987). The test line effects are assumed random, while the check variety effects are considered fixed. This may not be sensible or justifiable for most trials and can lead to inconsistent comparisons between check varieties and test lines. Given the large amount of replication afforded to check varieties there will be very little shrinkage irrespective of the realised heritability.

In the following we assume that the variety effects (including both check, replicated and unreplicated lines) is random. We consider an initial analysis with spatial correlation in one direction and present three further spatial models for comparison. The ASReml input file is presented below.

```
Tullibigeal trial
  linenum
  yield
  weed
  column 10
  row 67
  variety 532 # testlines 1:525, check lines 526:532
tulli.asd !skip 1 !DOPART 1
!PART 1 # AR1 x I
y ~ mu weed mv !r variety
1 2
67 row AR1 0.1
10 column I 0

!PART 2 # AR1 x AR1
y ~ mu weed mv !r variety
1 2
67 row AR1 0.1
10 column AR1 0.1

!PART 3 # AR1 x AR1 + column trend
y ~ mu weed pol(column,-1) mv !r variety
1 2
67 row AR1 0.1
10 column AR1 0.1

!PART 4 # AR1 x AR1 + Nugget + column trend
y ~ mu weed pol(column,-1) mv !r variety units
1 2
67 row AR1 0.1
10 column AR1 0.1
predict var
```

The data fields represent the factors `variety`, `row` and `column`, a covariate `weed` and the plot yield (`yield`). There are three parts in the ASReml file. We begin with the one-dimensional spatial model, which assumes the variance model for the plot effects within columns is described by a first order autoregressive process. The abbreviated output file is presented below.

1	LogL=-4280.75	S2= 0.12850E+06	666 df	0.1000	1.000	0.1000
2	LogL=-4268.57	S2= 0.12138E+06	666 df	0.1516	1.000	0.1798


```

3 LogL=-4255.89      S2= 0.10968E+06      666 df      0.2977      1.000      0.2980
4 LogL=-4243.76      S2= 88033.           666 df      0.7398      1.000      0.4939
5 LogL=-4240.59      S2= 84420.           666 df      0.9125      1.000      0.6016
6 LogL=-4240.01      S2= 85617.           666 df      0.9344      1.000      0.6428
7 LogL=-4239.91      S2= 86032.           666 df      0.9474      1.000      0.6596
8 LogL=-4239.88      S2= 86189.           666 df      0.9540      1.000      0.6668
9 LogL=-4239.88      S2= 86253.           666 df      0.9571      1.000      0.6700
10 LogL=-4239.88     S2= 86280.           666 df      0.9585      1.000      0.6714
Final parameter values          0.95918      1.0000      0.67205

Source          Model terms      Gamma      Component      Comp/SE      % C
variety          532      532      0.959184      82758.6      8.98      0 P
Variance          670      666      1.00000      86280.2      9.12      0 P
Residual          AR=AutoR      67      0.672052      0.672052      16.04      1 U

Analysis of Variance          DF      F-incr      F-adj
7 mu          1      9799.18      9336.79
3 weed          1      109.33      109.33

```

The iterative sequence converged, the REML estimate of the autoregressive parameter indicating substantial within column heterogeneity.

The abbreviated output from the two-dimensional AR1×AR1 spatial model is now presented.

```

1 LogL=-4277.99      S2= 0.12850E+06      666 df
2 LogL=-4266.13      S2= 0.12097E+06      666 df
3 LogL=-4253.05      S2= 0.10777E+06      666 df
4 LogL=-4238.72      S2= 83156.           666 df
5 LogL=-4234.53      S2= 79868.           666 df
6 LogL=-4233.78      S2= 82024.           666 df
7 LogL=-4233.67      S2= 82725.           666 df
8 LogL=-4233.65      S2= 82975.           666 df
9 LogL=-4233.65      S2= 83065.           666 df
10 LogL=-4233.65     S2= 83100.           666 df

Source          Model terms      Gamma      Component      Comp/SE      % C
variety          532      532      1.06038      88117.5      9.92      0 P
Variance          670      666      1.00000      83100.1      8.90      0 P
Residual          AR=AutoR      67      0.685387      0.685387      16.65      0 U
Residual          AR=AutoR      10      0.285909      0.285909      3.87      0 U

Analysis of Variance          DF      F-incr      F-adj
7 mu          1      6248.65      6117.37
3 weed          1      85.84      85.84

```

The change in REML log-likelihood is significant ($\chi_1^2 = 12.46, p < .001$) with

the inclusion of the autoregressive parameter for columns. Figure 15.6 presents the sample variogram of the residuals for the AR1×AR1 model. There is an indication that a linear drift from column 1 to column 10 is present. We include a linear regression coefficient `pol(column,-1)` in the model to account for this. Note we use the '-1' option in the `pol` term to exclude the overall constant in the regression, as it is already fitted. The linear regression of column number on yield is significant ($t = -2.96$). The sample variogram (15.7) is more satisfactory, though interpretation of variograms is often difficult, particularly for unreplicated trials. This is an issue for further research.

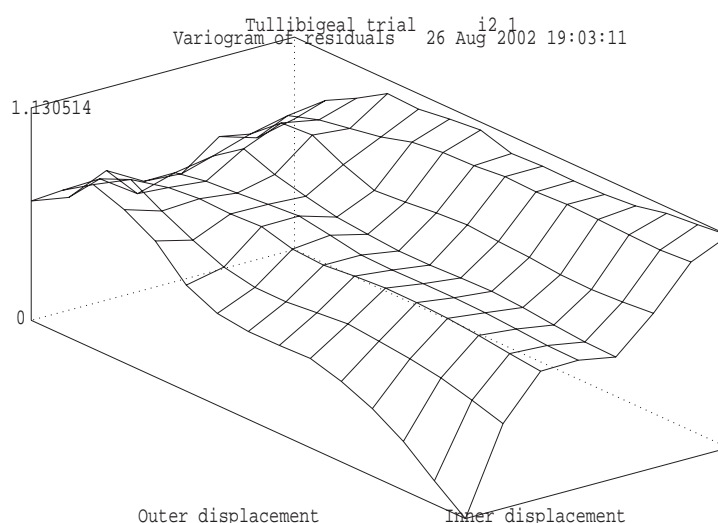


Figure 15.6 Sample variogram of the AR1×AR1 model for the Tullibigeal data

The abbreviated output for this model and the final model in which a nugget effect has been included is presented below.

```
#AR1xAR1 + pol(column,-1)
 1 LogL=-4270.99   S2= 0.12730E+06   665 df
 2 LogL=-4258.95   S2= 0.11961E+06   665 df
 3 LogL=-4245.27   S2= 0.10545E+06   665 df
 4 LogL=-4229.50   S2=  78387.         665 df
 5 LogL=-4226.02   S2=  75375.         665 df
 6 LogL=-4225.64   S2=  77373.         665 df
 7 LogL=-4225.60   S2=  77710.         665 df
 8 LogL=-4225.60   S2=  77786.         665 df
 9 LogL=-4225.60   S2=  77806.         665 df
```

Source	Model terms	Gamma	Component	Comp/SE	% C
--------	-------------	-------	-----------	---------	-----

```

variety          532   532   1.14370      88986.3      9.91   0 P
Variance         670   665   1.00000      77806.0      8.79   0 P
Residual        AR=AutoR   67  0.671436      0.671436     15.66   0 U
Residual        AR=AutoR   10  0.266088      0.266088      3.53   0 U

Analysis of Variance                DF    F-incr    F-adj
  7 mu                            1    7082.49   6726.77
  3 weed                            1      91.96    70.00
  8 pol(column,-1)                  1       8.74     8.74

#
#AR1xAR1 + units + pol(column,-1)
#
  1 LogL=-4272.74    S2= 0.11683E+06    665 df    : 1 components constrained
  2 LogL=-4266.07    S2= 50207.          665 df    : 1 components constrained
  3 LogL=-4228.96    S2= 76724.          665 df
  4 LogL=-4220.63    S2= 55858.          665 df
  5 LogL=-4220.19    S2= 54431.          665 df
  6 LogL=-4220.18    S2= 54732.          665 df
  7 LogL=-4220.18    S2= 54717.          665 df
  8 LogL=-4220.18    S2= 54715.          665 df

Source          Model terms      Gamma      Component      Comp/SE      % C
variety         532   532   1.34824      73769.0        7.08   0 P
units           670   670   0.556400     30443.6        3.77   0 P
Variance        670   665   1.00000     54715.2        5.15   0 P
Residual        AR=AutoR   67  0.837503     0.837503       18.67   0 U
Residual        AR=AutoR   10  0.375382     0.375382        3.26   0 U

Analysis of Variance                DF    F-incr    F-adj
  7 mu                            1    4241.39   4259.90
  3 weed                            1      86.39    72.73
  8 pol(column,-1)                  1       4.84     4.84

```

The increase in REML log-likelihood is significant. The predicted means for the varieties can be produced and printed in the .pvs file as shown below

```

Warning: mv_estimates      is ignored for prediction
Warning: units             is ignored for prediction

----- 1 -----
column          evaluated at      5.5000
weed            is evaluated at average value of      0.4597
Predicted values of yield

variety          Predicted_Value Standard_Error Ecode
  1.0000          2917.1782      179.2881 E
  2.0000          2957.7405      178.7688 E
  3.0000          2872.7615      176.9880 E

```

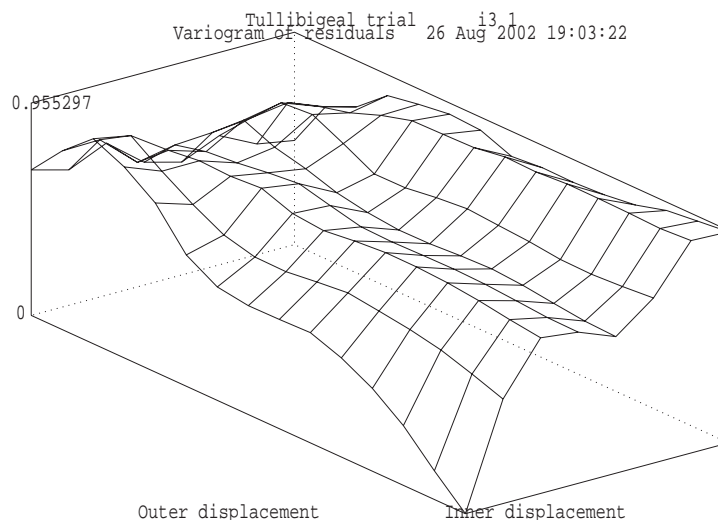


Figure 15.7: Sample variogram of the $AR1 \times AR1 + pol(column, -1)$ model for the Tullibigeal data

4.0000	2986.4725	178.7424 E
522.0000	2784.7683	179.1541 E
523.0000	2904.9421	179.5383 E
524.0000	2740.0330	178.8465 E
525.0000	2669.9565	179.2444 E
526.0000	2385.9806	44.2159 E
527.0000	2697.0670	133.4406 E
528.0000	2727.0324	112.2650 E
529.0000	2699.8243	103.9062 E
530.0000	3010.3907	112.3080 E
531.0000	3020.0720	112.2553 E
532.0000	3067.4479	112.6645 E
SED: Overall Standard Error of Difference		245.8

Note that the (replicated) check lines have lower SE than the (unreplicated) test lines. There will also be large differences in SEDs. Rather than obtaining the large table of all SEDs, you could do the prediction in parts

```
predict var 1:525 column 5.5
predict var 526:532 column 5.5 !SED
```

to examine the matrix of pairwise prediction errors of variety differences.

15.8 Paired Case-Control study - Rice

This data is concerned with an experiment conducted to investigate the tolerance of rice varieties to attack by the larvae of bloodworms. The data have been kindly provided by Dr. Mark Stevens, Yanco Agricultural Institute. A full description of the experiment is given by Stevens *et al.* (1999). Bloodworms are a significant pest of rice in the Murray and Murrumbidgee irrigation areas where they can cause poor establishment and substantial yield loss.

The experiment commenced with the transplanting of rice seedlings into trays. Each tray contained 32 seedlings and the trays were paired so that a control tray (no bloodworms) and a treated tray (bloodworms added) were grown in a controlled environment room for the duration of the experiment. At the end of this time rice plants were carefully extracted, the root system washed and root area determined for the tray using an image analysis system described by Stevens *et al.* (1999). Two pairs of trays, each pair corresponding to a different variety, were included in each run. A new batch of bloodworm larvae was used for each run. A total of 44 varieties was investigated with three replicates of each. Unfortunately the variety concurrence within runs was less than optimal. Eight varieties occurred with only one other variety, 22 with two other varieties and the remaining 14 with three different varieties.

In the next three sections we present an exhaustive analysis of these data using equivalent univariate and multivariate techniques. It is convenient to use two data files one for each approach. The univariate data file consists of factors `pair`, `run`, `variety`, `tmt`, `unit` and variate `rootwt`. The factor `unit` labels the individual trays, `pair` labels pairs of trays (to which varieties are allocated) and `tmt` is the two level bloodworm treatment factor (control/treated). The multivariate data file consists of factors `variety` and `run` and variates for root weight of both the control and exposed treatments (labelled `yc` and `ye` respectively).

Figure 15.8 presents a plot of the treated and the control root area (on the square root scale) for each variety. There is a strong dependence between the treated and control root area, which is not surprising. The aim of the experiment was to determine the tolerance of varieties to bloodworms thence identify the most tolerant varieties. The definition of tolerance should allow for the fact that varieties differ in their inherent seedling vigour (Figure 15.8). The original approach of the scientist was to regress the treated root area against the control root area and define the index of vigour as the residual from this regression. This approach is clearly inefficient since there is error in both variables. We seek to

determine an index of tolerance from the joint analysis of treated and control root area.

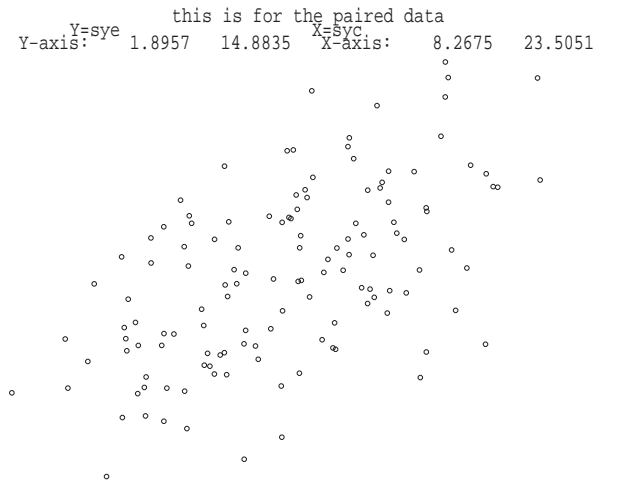


Figure 15.8: Rice bloodworm data: Plot of square root of root weight for treated versus control

Standard analysis

Preliminary analyses indicated variance heterogeneity so that subsequent analyses were conducted on the square root scale. The allocation of bloodworm treatments within varieties and varieties within runs defines a nested block structure of the form

```
run/variety/tmt = run + run.variety + run.variety.tmt
                (= run + pair + pair.tmt )
                (= run + run.variety + units )
```

There is an additional blocking term, however, due to the fact that the bloodworms within a run are derived from the same batch of larvae whereas between runs the bloodworms come from different sources. This defines a block structure of the form

```
run/tmt/variety = run + run.tmt + run.tmt.variety
                 (= run + run.tmt + pair.tmt )
```

Combining the two provides the full block structure for the design, namely

```
run + run.variety + run.tmt + run.tmt.variety
= run + run.variety + run.tmt + units
= run + pair + run.tmt + pair.tmt
```

In line with the aims of the experiment the treatment structure comprises variety and treatment main effects and treatment by variety interactions. In the traditional approach the terms in the block structure are regarded as random and the treatment terms as fixed. The choice of treatment terms as fixed or random depends largely on the aims of the experiment. The aim of this example is to select the "best" varieties. The definition of best is somewhat more complex since it does not involve the single trait $\text{sqrt}(\text{rootwt})$ but rather two traits, namely $\text{sqrt}(\text{rootwt})$ in the presence/absence of bloodworms. Thus to minimise selection bias the variety main effects and thence the `tmt.variety` interactions are taken as random. The main effect of treatment is fitted as fixed to allow for the likely scenario that rather than a single population of treatment by variety effects there are in fact two populations (control and treated) with a different mean for each. There is evidence of this prior to analysis with the large difference in mean $\text{sqrt}(\text{rootwt})$ for the two groups (14.93 and 8.23 for control and treated respectively). The inclusion of `tmt` as a fixed effect ensures that BLUPs of `tmt.variety` effects are shrunk to the correct mean (treatment means rather than an overall mean).

The model for the data is given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\tau} + \mathbf{Z}_1\mathbf{u}_1 + \mathbf{Z}_2\mathbf{u}_2 + \mathbf{Z}_3\mathbf{u}_3 + \mathbf{Z}_4\mathbf{u}_4 + \mathbf{Z}_5\mathbf{u}_5 + \mathbf{e} \quad (15.7)$$

where \mathbf{y} is a vector of length $n = 264$ containing the $\text{sqrt}(\text{rootwt})$ values, $\boldsymbol{\tau}$ corresponds to a constant term and the fixed treatment contrast and $\mathbf{u}_1 \dots \mathbf{u}_5$ correspond to random variety, treatment by variety, run, treatment by run and variety by run effects. The random effects and error are assumed to be independent Gaussian variables with zero means and variance structures $\text{var}(\mathbf{u}_i) = \sigma_i^2 \mathbf{I}_{b_i}$ (where b_i is the length of \mathbf{u}_i , $i = 1 \dots 5$) and $\text{var}(\mathbf{e}) = \sigma^2 \mathbf{I}_n$.

The ASReml code for this analysis is given by

```
Bloodworm data Dr M Stevens
pair 132
rootwt
```

```

run 66
tmt 2 !A
id
variety 44 !A
worm.txt !skip 1 !DOPART 1
!PART 1
sqrt(rootwt) ~ mu tmt !r variety variety.tmt run pair run.tmt
0 0 0
!PART 2
sqrt(rootwt) ~ mu tmt !r variety tmt.variety run pair tmt.run,
uni(tmt,2)
0 0 2
tmt.variety 2
2 0 DIAG .1 .1 !GU
44 0 0
tmt.run 2
2 0 DIAG .1 .1 !GU
66 0 0

```

The two parts in the input file define the two univariate analyses we will conduct. We consider the results from the analysis defined in PART 1 first. A portion of the output file is presented below.

1	LogL=-369.089	S2=	4.1533	262	df
2	LogL=-357.871	S2=	2.6834	262	df
3	LogL=-349.520	S2=	1.7986	262	df
4	LogL=-345.739	S2=	1.3719	262	df
5	LogL=-345.306	S2=	1.3216	262	df
6	LogL=-345.267	S2=	1.3155	262	df
7	LogL=-345.264	S2=	1.3149	262	df
8	LogL=-345.263	S2=	1.3149	262	df

Source	Model	terms	Gamma	Component	Comp/SE	% C
variety	44	44	1.80947	2.37920	3.01	0 P
run	66	66	0.244243	0.321144	0.59	0 P
variety.tmt	88	88	0.374220	0.492047	1.78	0 P
pair	132	132	0.742328	0.976057	2.51	0 P
run.tmt	132	132	1.32973	1.74841	3.65	0 P
Variance	264	262	1.00000	1.31486	4.42	0 P

Analysis of Variance		DF	F-incr
7	mu	1	1484.96
4	tmt	1	469.35

The estimated variance components from this analysis are given in column (a) of table 15.9. The variance component for the `variety` main effects is large. There

Table 15.9: Estimated variance components from univariate analyses of bloodworm data. (a) Model with homogeneous variance for all terms and (b) Model with heterogeneous variance for interactions involving `tmt`

source	(a)	(b)	
		control	treated
variety	2.378	2.334	
tmt.variety	0.492	1.505	-0.372
run	0.321	0.319	
tmt.run	1.748	1.388	2.223
variety.run (pair)	0.976	0.987	
tmt.pair	1.315	1.156	1.359
REML log-likelihood	-345.256	-343.22	

is evidence of `tmt.variety` interactions so we may expect some discrimination between varieties in terms of tolerance to bloodworms.

Given the large difference ($p < 0.001$) between `tmt` means we may wish to allow for heterogeneity of variance associated with `tmt`. Thus we fit a separate `variety` variance for each level of `tmt` so that instead of assuming $\text{var}(\mathbf{u}_2) = \sigma_2^2 \mathbf{I}_{88}$ we assume

$$\text{var}(\mathbf{u}_2) = \begin{bmatrix} \sigma_{2c}^2 & 0 \\ 0 & \sigma_{2t}^2 \end{bmatrix} \otimes \mathbf{I}_{44}$$

where σ_{2c}^2 and σ_{2t}^2 are the `tmt.variety` interaction variances for control and treated respectively. This model can be achieved using a diagonal variance structure for the treatment part of the interaction. We also fit a separate `run` variance for each level of `tmt` and heterogeneity at the residual level, by including the `uni(tmt,2)` term. We have chosen level 2 of `tmt` as we expect more variation for the exposed treatment and thus the extra variance component for this term should be positive. Had we mistakenly specified level 1 then `ASReml` would have estimated a negative component by setting the `!GU` option for this term. The portion of the `ASReml` output for this analysis is given below.

```

1 LogL=-369.991      S2=  4.0377      262 df
2 LogL=-357.468      S2=  2.5831      262 df
3 LogL=-348.310      S2=  1.6917      262 df      :   1 components constrained

```

4	LogL=-344.027	S2= 1.2176	262 df	:	1 components constrained
5	LogL=-343.573	S2= 1.1605	262 df	:	1 components constrained
6	LogL=-343.428	S2= 1.1498	262 df	:	1 components constrained
7	LogL=-343.234	S2= 1.1531	262 df		
8	LogL=-343.228	S2= 1.1572	262 df		
9	LogL=-343.228	S2= 1.1563	262 df		

Source	Model	terms	Gamma	Component	Comp/SE	% C
variety	44	44	2.01903	2.33451	3.01	0 P
run	66	66	0.276045	0.319178	0.59	0 P
pair	132	132	0.853941	0.987372	2.59	0 P
uni(tmt,2)	264	264	0.176158	0.203684	0.32	0 P
Variance	264	262	1.00000	1.15625	2.77	0 P
tmt.variety	DIAGonal	1	1.30142	1.50477	2.26	0 U
tmt.variety	DIAGonal	2	-0.321901	-0.372199	-0.82	0 U
tmt.run	DIAGonal	1	1.20098	1.38864	2.18	0 U
tmt.run	DIAGonal	2	1.92457	2.22530	3.07	0 U

Analysis of Variance	DF	F-incr
7 mu	1	1276.91
4 tmt	1	448.86

The estimated variance components from this analysis are given in column (b) of table 15.9. There is no significant variance heterogeneity at the residual or `tmt.run` level. This indicates that the square root transformation of the data has successfully stabilised the error variance. There is, however, significant variance heterogeneity for `tmt.variety` interactions with the variance being much greater for the control group. This reflects the fact that in the absence of bloodworms the potential maximum root area is greater. Note that the `tmt.variety` interaction variance for the treated group is negative. The negative component is meaningful (and in fact necessary and obtained by use of the `!GU` option) in this context since it should be considered as part of the variance structure for the combined variety main effects and treatment by variety interactions. That is,

$$\text{var}(\mathbf{1}_2 \otimes \mathbf{u}_1 + \mathbf{u}_2) = \begin{bmatrix} \sigma_1^2 + \sigma_{2c}^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma_{2t}^2 \end{bmatrix} \otimes \mathbf{I}_{44} \quad (15.8)$$

Using the estimates from table 15.9 this structure is estimated as

$$\begin{bmatrix} 3.84 & 2.33 \\ 2.33 & 1.96 \end{bmatrix} \otimes \mathbf{I}_{44}$$

Thus the variance of the variety effects in the control group (also known as the genetic variance for this group) is 3.84. The genetic variance for the treated group is much lower (1.96). The genetic correlation is $2.33/\sqrt{3.84 * 1.96} = 0.85$ which is strong, supporting earlier indications of the dependence between the treated and control root area (Figure 15.8).

Table 15.10 Equivalence of random effects in bivariate and univariate analyses

effects	bivariate (model 15.10)	univariate (model 15.7)
trait.variety	\mathbf{u}_v	$\mathbf{1}_2 \otimes \mathbf{u}_1 + \mathbf{u}_2$
trait.run	\mathbf{u}_r	$\mathbf{1}_2 \otimes \mathbf{u}_3 + \mathbf{u}_4$
trait.pair	\mathbf{e}^*	$\mathbf{1}_2 \otimes \mathbf{u}_5 + \mathbf{e}$

A multivariate approach

In this simple case in which the variance heterogeneity is associated with the two level factor `tmt`, the analysis is equivalent to a bivariate analysis in which the two traits correspond to the two levels of `tmt`, namely `sqrt(rootwt)` for control and treated. The model for each trait is given by

$$\mathbf{y}_j = \mathbf{X}\boldsymbol{\tau}_j + \mathbf{Z}_v\mathbf{u}_{v_j} + \mathbf{Z}_r\mathbf{u}_{r_j} + \mathbf{e}_j \quad (j = c, t) \quad (15.9)$$

where \mathbf{y}_j is a vector of length $n = 132$ containing the `sqrtroot` values for variate j ($j = c$ for control and $j = t$ for treated), $\boldsymbol{\tau}_j$ corresponds to a constant term and \mathbf{u}_{v_j} and \mathbf{u}_{r_j} correspond to random variety and run effects. The design matrices are the same for both traits. The random effects and error are assumed to be independent Gaussian variables with zero means and variance structures $\text{var}(\mathbf{u}_{v_j}) = \sigma_{v_j}^2 \mathbf{I}_{44}$, $\text{var}(\mathbf{u}_{r_j}) = \sigma_{r_j}^2 \mathbf{I}_{66}$ and $\text{var}(\mathbf{e}_j) = \sigma_j^2 \mathbf{I}_{132}$. The bivariate model can be written as a direct extension of (15.9), namely

$$\mathbf{y} = (\mathbf{I}_2 \otimes \mathbf{X}) \boldsymbol{\tau} + (\mathbf{I}_2 \otimes \mathbf{Z}_v) \mathbf{u}_v + (\mathbf{I}_2 \otimes \mathbf{Z}_r) \mathbf{u}_r + \mathbf{e}^* \quad (15.10)$$

where $\mathbf{y} = (\mathbf{y}'_c, \mathbf{y}'_t)'$, $\mathbf{u}_v = (\mathbf{u}'_{v_c}, \mathbf{u}'_{v_t})'$, $\mathbf{u}_r = (\mathbf{u}'_{r_c}, \mathbf{u}'_{r_t})'$ and $\mathbf{e}^* = (\mathbf{e}'_c, \mathbf{e}'_t)'$.

There is an equivalence between the effects in this bivariate model and the univariate model of (15.7). The variety effects for each trait (\mathbf{u}_v in the bivariate model) are partitioned in (15.7) into variety main effects and `tmt.variety` interactions so that $\mathbf{u}_v = \mathbf{1}_2 \otimes \mathbf{u}_1 + \mathbf{u}_2$. There is a similar partitioning for the run effects and the errors (see table 15.10).

In addition to the assumptions in the models for individual traits (15.9) the bivariate analysis involves the assumptions $\text{cov}(\mathbf{u}_{v_c}) \mathbf{u}'_{v_t} = \sigma_{v_{ct}} \mathbf{I}_{44}$, $\text{cov}(\mathbf{u}_{r_c}) \mathbf{u}'_{r_t} = \sigma_{r_{ct}} \mathbf{I}_{66}$ and $\text{cov}(\mathbf{e}_c) \mathbf{e}'_t = \sigma_{ct} \mathbf{I}_{132}$. Thus random effects and errors are correlated between traits. So, for example, the variance matrix for the variety effects for

each trait is given by

$$\text{var}(\mathbf{u}_v) = \begin{bmatrix} \sigma_{v_c}^2 & \sigma_{v_{ct}} \\ \sigma_{v_{ct}} & \sigma_{v_t}^2 \end{bmatrix} \otimes \mathbf{I}_{44}$$

This unstructured form for `trait.variety` in the bivariate analysis is equivalent to the `variety` main effect plus heterogeneous `tmt.variety` interaction variance structure (15.8) in the univariate analysis. Similarly the unstructured form for `trait.run` is equivalent to the `run` main effect plus heterogeneous `tmt.run` interaction variance structure. The unstructured form for the errors (`trait.pair`) in the bivariate analysis is equivalent to the `pair` plus heterogeneous error (`tmt.pair`) variance in the univariate analysis.

This bivariate analysis is achieved in ASReml as follows, noting that the `tmt` factor here is equivalent to traits

```
this is for the paired data
id
pair 132
run 66
variety 44 !A
yc ye
wormm.asd !skip 1 !X syc !Y sye
sqrt(yc) sqrt(ye) ~ Trait !r Tr.variety Tr.run
1 2 2
132 !S2==1
Tr 0 US 2.21 1.1 2.427
Tr.variety 2
2 0 US 1.401 1 1.477
44 0 0
Tr.run 2
2 0 US .79 .5 2.887
66 0 0
predict variety
```

A portion of the output from this analysis is presented below.

1	LogL=-354.998	S2= 1.0000	262 df
2	LogL=-350.056	S2= 1.0000	262 df
3	LogL=-345.557	S2= 1.0000	262 df
4	LogL=-343.386	S2= 1.0000	262 df
5	LogL=-343.226	S2= 1.0000	262 df
6	LogL=-343.220	S2= 1.0000	262 df
7	LogL=-343.220	S2= 1.0000	262 df
8	LogL=-343.220	S2= 1.0000	262 df

```

Source          Model terms      Gamma      Component  Comp/SE  % C
Residual       UnStruct  1  2.14373    2.14373    4.44  0 U
Residual       UnStruct  1  0.987401   0.987401   2.59  0 U
Residual       UnStruct  2  2.34751    2.34751    4.62  0 U
Tr.variety     UnStruct  1  3.83959    3.83959    3.47  0 U
Tr.variety     UnStruct  1  2.33394    2.33394    3.01  0 U
Tr.variety     UnStruct  2  1.96173    1.96173    2.69  0 U
Tr.run         UnStruct  1  1.70788    1.70788    2.62  0 U
Tr.run         UnStruct  1  0.319145   0.319145   0.59  0 U
Tr.run         UnStruct  2  2.54326    2.54326    3.20  0 U
Covariance/Variance/Correlation Matrix UnStructured
  2.144    0.4402
0.9874    2.348
Covariance/Variance/Correlation Matrix UnStructured
  3.840    0.8504
  2.334    1.962
Covariance/Variance/Correlation Matrix UnStructured
  1.708    0.1531
0.3191    2.543

```

The resultant REML log-likelihood is identical to that of the heterogeneous univariate analysis (column (b) of table 15.9). The estimated variance parameters are given in Table 15.11.

Predicted variety means are produced in the `.pvs`. These are used in the following section on interpretation of results. A portion of the file is presented below. There is a wide range in SED reflecting the imbalance of the variety concurrence within runs.

```
Assuming Power transformation was (Y+ 0.000)^ 0.500
```

Table 15.11: Estimated variance parameters from bivariate analysis of bloodworm data

source	control variance	treated variance	covariance
us(trait).variety	3.84	1.96	2.33
us(trait).run	1.71	2.54	0.32
us(trait).pair	2.14	2.35	0.99

```

run          is ignored in the prediction (except where specifically included

Trait       variety      Power_value  Stand_Error  Ecode  Retransformed  approx_SE
sqrt(yc)    AliCombo      14.9532     0.9181 E    223.5982  27.4571
sqrt(ye)    AliCombo      7.9941      0.7993 E    63.9054   12.7790
sqrt(yc)    Bluebelle     13.1033     0.9310 E    171.6969  24.3980
sqrt(ye)    Bluebelle     6.6299      0.8062 E    43.9559   10.6901
sqrt(yc)    C22           16.6679     0.9181 E    277.8192  30.6057
sqrt(ye)    C22           8.9543      0.7993 E    80.1798   14.3140
sqrt(yc)    Chiyohikari  17.3393     0.9301 E    300.6510  32.2538
sqrt(ye)    Chiyohikari  9.6433      0.8057 E    92.9927   15.5393
.           .             .           .           .         .
sqrt(yc)    YRK1         15.1859     0.9549 E    230.6103  29.0012
sqrt(ye)    YRK1         8.3356      0.8190 E    69.4817   13.6534
sqrt(yc)    YRK3         13.3057     0.9549 E    177.0428  25.4106
sqrt(ye)    YRK3         8.1133      0.8190 E    65.8264   13.2894
SED: Overall Standard Error of Difference  1.215

```

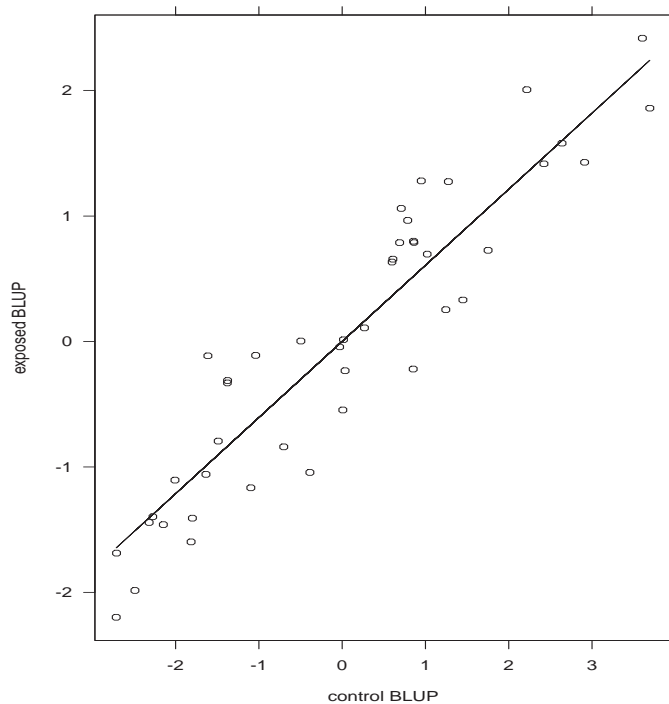


Figure 15.9 BLUPs for treated for each variety plotted against BLUPs for control

Interpretation of results

Recall that the researcher is interested in varietal tolerance to bloodworms. This could be defined in various ways. One option is to consider the regression implicit in the variance structure for the trait by variety effects. The variance structure can arise from a regression of treated variety effects on control effects, namely

$$\mathbf{u}_{vt} = \beta \mathbf{u}_{vc} + \boldsymbol{\epsilon}$$

where the slope $\beta = \sigma_{vct} / \sigma_{vc}^2$. Tolerance can be defined in terms of the deviations from regression, $\boldsymbol{\epsilon}$. Varieties with large positive deviations have greatest tolerance to bloodworms. Note that this is similar to the researcher's original intentions except that the regression has been conducted at the genotypic rather than the phenotypic level. In Figure 15.9 the BLUPs for treated have been plotted against the BLUPs for control for each variety and the fitted regression line (slope = 0.61) has been drawn. Varieties with large positive deviations from the regression line include YRK3, Calrose, HR19 and WC1403.

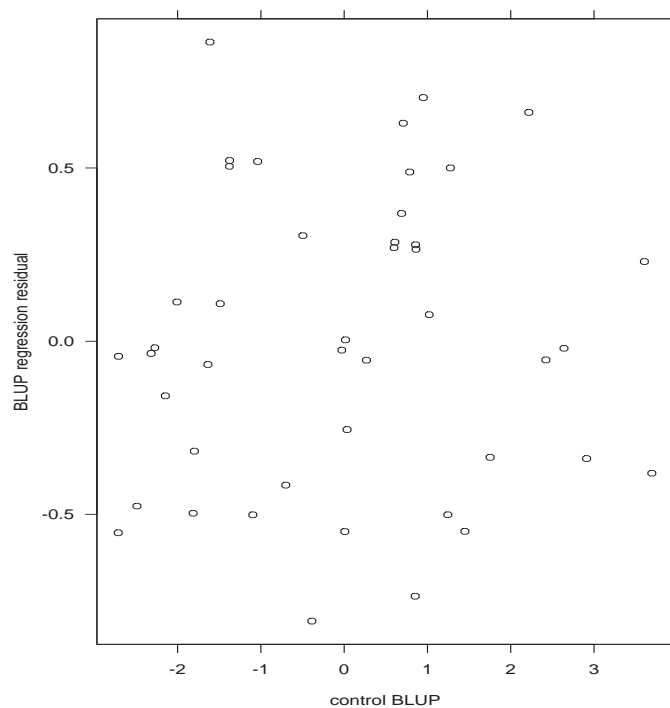


Figure 15.10: Estimated deviations from regression of treated on control for each variety plotted against estimate for control

An alternative definition of tolerance is the simple difference between treated and control BLUPs for each variety, namely $\delta = \mathbf{u}_{v_c} - \mathbf{u}_{v_t}$. Unless $\beta = 1$ the two measures ϵ and δ have very different interpretations. The key difference is that ϵ is a measure which is *independent* of inherent vigour whereas δ is not. To see this consider

$$\begin{aligned} \text{cov}(\epsilon) \mathbf{u}'_{v_c} &= \text{cov}(\mathbf{u}_{v_t} - \beta \mathbf{u}_{v_c}) \mathbf{u}'_{v_c} \\ &= \left(\sigma_{v_{ct}} - \frac{\sigma_{v_{ct}}}{\sigma_{v_c}^2} \sigma_{v_c}^2 \right) \mathbf{I}_{44} \\ &= \mathbf{0} \end{aligned}$$

whereas

$$\begin{aligned} \text{cov}(\delta) \mathbf{u}'_{v_c} &= \text{cov}(\mathbf{u}_{v_c} - \mathbf{u}_{v_t}) \mathbf{u}'_{v_c} \\ &= (\sigma_{v_c}^2 - \sigma_{v_{ct}}) \mathbf{I}_{44} \end{aligned}$$

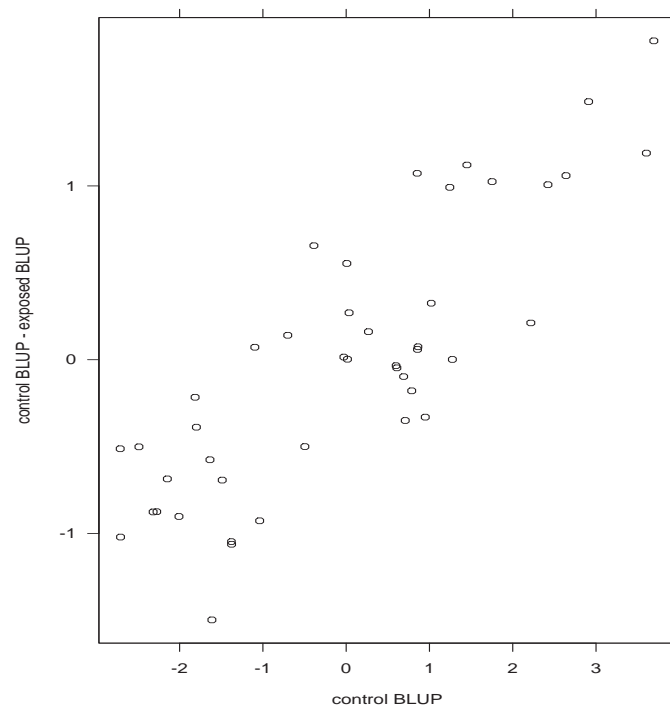


Figure 15.11: Estimated difference between control and treated for each variety plotted against estimate for control

The independence of ϵ and \mathbf{u}_{v_c} and dependence between δ and \mathbf{u}_{v_c} is clearly illustrated in Figures 15.10 and 15.11. In this example the two measures have provided very different rankings of the varieties. The choice of tolerance measure depends on the aim of the experiment. In this experiment the aim was to identify tolerance which is independent of inherent vigour so the deviations from regression measure is preferred.

15.9 Balanced longitudinal data - Random coefficients and cubic smoothing splines - Oranges

We now illustrate the use of random coefficients and cubic smoothing splines for the analysis of balanced longitudinal data. The implementation of cubic smoothing splines in `ASReml` was originally based on the mixed model formulation presented by Verbyla *et al.* (1999). More recently the technology has been enhanced so that the user can specify knot points; in the original approach the knot points were taken to be the ordered set of unique values of the explanatory variable. The specification of knot points is particularly useful if the number of unique values in the explanatory variable is large, or if units are measured at different times.

The data we use was originally reported by Draper and Smith (1998, ex24N, p559) and has recently been reanalysed by Pinheiro and Bates (2000, p338). The data are displayed in Figure 15.12 and are the trunk circumferences (in millimetres) of each of 5 trees taken at 7 times. All trees were measured at the same time so that the data are balanced. The aim of the study is unclear, though, both previous analyses involved modelling the overall ‘growth’ curve, accounting for the obvious variation in both level and shape between trees. Pinheiro and Bates (2000) used a nonlinear mixed effects modelling approach, in which they modelled the growth curves by a three parameter logistic function of age, given by

$$y = \frac{\phi_1}{1 + \exp[-(x - \phi_2)/\phi_3]} \quad (15.11)$$

where y is the trunk circumference, x is the tree age in days since December 31 1968, ϕ_1 is the asymptotic height, ϕ_2 is the inflection point or the time at which the tree reaches $0.5\phi_1$, ϕ_3 is the time elapsed between trees reaching half and about $3/4$ of ϕ_1 .

The datafile consists of 5 columns viz, `Tree`, a factor with 5 levels, `x`, tree age in days since 31st December 1968, `circ` the trunk circumference and `season`. The last column `season` was added after noting that tree age spans several years and if



Figure 15.12 Trellis plot of trunk circumference for each tree

converted to day of year, measurements were taken in either **Spring** (April/May) or **Autumn** (September/October).

First we demonstrate the fitting of a cubic spline in **ASReml** by restricting the dataset to tree 1 only. The model includes the intercept and linear regression of trunk circumference on x and an additional random term `spl(x,7)` which instructs **ASReml** to include a random term with a special design matrix with $7 - 2 = 5$ columns which relate to the vector, δ whose elements $\delta_i, i = 2, \dots, 6$ are the second differentials of the cubic spline at the knot points. The second differentials of a natural cubic spline are zero at the first and last knot points (Green and Silverman, 1994). The **ASReml** job is

```
this is the orange data, for tree 1
seq      # record number is not used
Tree 5
x        # 118  484  664 1004 1231 1372 1582
circ
season !L Spring Autumn
orange.asd !skip 1 !filter 2 !select 1
!SPLINE spl(x,7) 118  484  664 1004 1231 1372 1582
!PVAL x 150 200:1500
circ ~ mu x !r spl(x,7)
predict x
```

Note that the data for tree 1 has been selected by use of the `!filter` and `!select` qualifiers. Also note the use of `!PVAL` so that the spline curve is properly predicted at the additional nominated points. These additional data points are required for ASReml to form the design matrix to properly interpolate the cubic smoothing spline between knot points in the prediction process. Since the spline knot points are specifically nominated in the `!SPLINE` line, these extra points have no effect on the analysis run time. The `!SPLINE` line does not modify the analysis in this example since it simply nominates the 7 ages in the data file. The same analysis would result if the `!SPLINE` line was omitted and `spl(x,7)` in the model was replaced with `spl(x)`.

An extract of the output file is presented below

```

1 LogL=-20.9043      S2= 48.470           5 df  0.1000      1.000
2 LogL=-20.9017      S2= 49.022           5 df  0.9266E-01  1.000
3 LogL=-20.8999      S2= 49.774           5 df  0.8356E-01  1.000
4 LogL=-20.8996      S2= 50.148           5 df  0.7937E-01  1.000
5 LogL=-20.8996      S2= 50.213           5 df  0.7866E-01  1.000
Final parameter values                0.78798E-01 1.0000

Source          Model terms      Gamma      Component      Comp/SE      % C
spl(x,7)         5          5  0.787982E-01  3.95671        0.40  0 P
Variance         7          5  1.00000      50.2133        1.32  0 P

Analysis of Variance                DF      F-incr      F-adj
7 mu                                1      1382.13     18.03
3 x                                  1        217.50     217.50
NOTICE: ASReml cannot yet determine the correct denominator DFs for
these F statistics. They will not exceed the Residual DF
which applies only if all components are KNOWN.

              Estimate      Standard Error      T-value      T-prev
3 x
1 -0.126873      0.552471E-02      -22.96
7 mu
2 -0.397376      5.75569           -0.07
6 spl(x,7)                5 effects fitted
Finished: 30 Sep 2002 20:24:53.752  LogL Converged

```

The REML estimate of the smoothing constant indicates that there is some non-linearity. The fitted cubic smoothing spline is presented in Figure 15.13. The fitted values were obtained from the `.pvs` file. The four points below the line were the spring measurements.

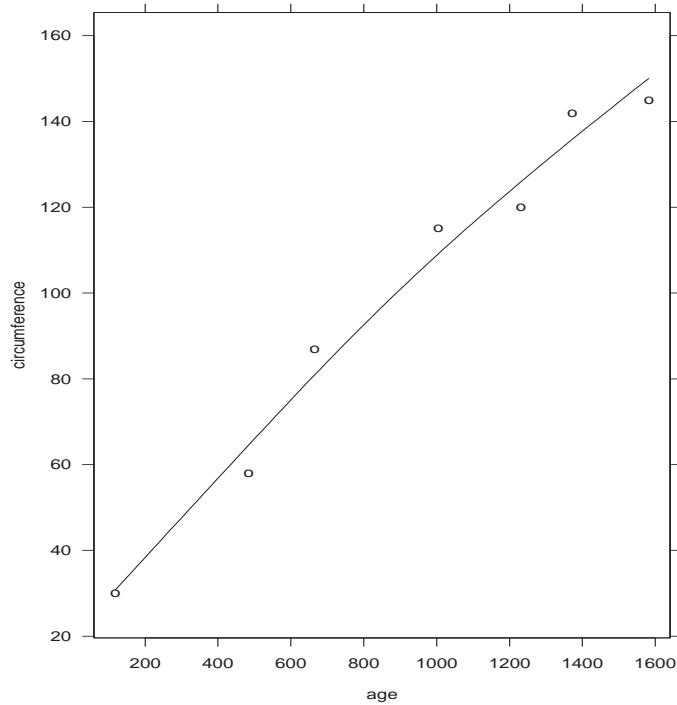


Figure 15.13 Fitted cubic smoothing spline for tree 1

We now consider the analysis of the full dataset. Following Verbyla *et al.* (1999) we consider the analysis of variance decomposition (see Table 15.12) which models the overall and individual curves.

An overall spline is fitted as well as tree deviation splines. We note however, that the intercept and slope for the tree deviation splines are assumed to be random effects. This is consistent with Verbyla *et al.* (1999). In this sense the tree deviation splines play a role in modelling the conditional curves for each tree and variance modelling. The intercept and slope for each tree are included as random coefficients (denoted by RC in Table 15.12). Thus, if $\mathbf{U}^{5 \times 2}$ is the matrix of intercepts (column 1) and slopes (column 2) for each tree, then we assume that

$$\text{var}(\text{vec}(\mathbf{U})) = \mathbf{\Sigma} \otimes \mathbf{I}_5$$

where $\mathbf{\Sigma}$ is a 2×2 symmetric positive definite matrix. Non smooth variation can be modelled at the overall mean (across trees) level and this is achieved in ASReml by inclusion of $\mathbf{fac}(\mathbf{x})$ as a random term.

Table 15.12 Orange data: AOV decomposition

stratum	decomposition	type	df or ne
constant	1	F	1
age	x	F	1
	spl(x,7)	R	5
	fac(x)	R	7
tree	tree	RC	5
	age.tree		
	x.tree	RC	5
	spl(x,7).tree	R	25
error		R	

An extract of the ASReml input file is presented below.

```

circ ~ mu x !r Tree 4.6 Tree.x .000094 spl(x,7) .1,
spl(x,7).Tree 2.3 fac(x) 13.9
0 0 1
Tree 2
2 0 US 4.6 .00001 .000094
5 0 0
predict x Tree !IGNORE fac(x)

```

We stress the importance of model building in these settings, where we generally commence with relatively simple variance models and update to more complex variance models if appropriate. Table 15.13 presents the sequence of fitted models we have used. Note that the REML log-likelihoods for models 1 and 2 are comparable and likewise for models 3 to 6. The REML log-likelihoods are not comparable between these groups due to the inclusion of the fixed `season` term in the second set of models.

We begin by modelling the variance matrix for the intercept and slope for each tree, Σ , as a diagonal matrix as there is no point including a covariance component between the intercept and slope if the variance component(s) for one (or

Table 15.13 Sequence of models fitted to the Orange data

term	model					
	1	2	3	4	5	6
tree	y	y	y	y	y	y
x.tree	y	y	y	y	y	y
(covariance)	n	n	n	n	n	y
spl(x,7)	y	y	y	y	n	y
tree.spl(x,7)	y	y	y	n	y	y
fac(x)	n	y	y	n	n	n
season	n	n	y	y	y	y
REML log-likelihood	-97.78	-94.07	-87.95	-91.22	-90.18	-87.43

both) is zero. Model 1 also does not include a non-smooth component at the overall level (that is, `fac(x)`). Abbreviated output is shown below.

```

11 LogL=-97.7788      S2= 6.3702      33 df

Source              Model  terms      Gamma      Component      Comp/SE      % C
Tree                5      5      4.78452      30.4782      1.24      0 P
Tree.x              5      5      0.938442E-04  0.597805E-03  1.41      0 P
spl(x,7)            5      5      100.403      639.587      1.55      0 P
spl(x,7).Tree       25     25      1.11475      7.10114      1.44      0 P
Variance            35     33      1.00000      6.37018      1.74      0 P

Analysis of Variance      DF      F-incr      F-adj
 8 mu                    1      47.06      43.77
 5 x                      1      95.00      95.00

```

A quick look suggests this is fine until we look at the predicted curves in Figure 15.14. The fit is unacceptable because the spline has picked up too much curvature, and suggests that there may be systematic non-smooth variation at the overall level. This can be formally examined by including the `fac(x)` term as a random effect. This increased the log-likelihood 3.71 ($P < 0.05$) with the `spl(x,7)` smoothing constants heading to the boundary. There is a possible explanation in the `season` factor. When this is added (Model 3) it has an F ratio

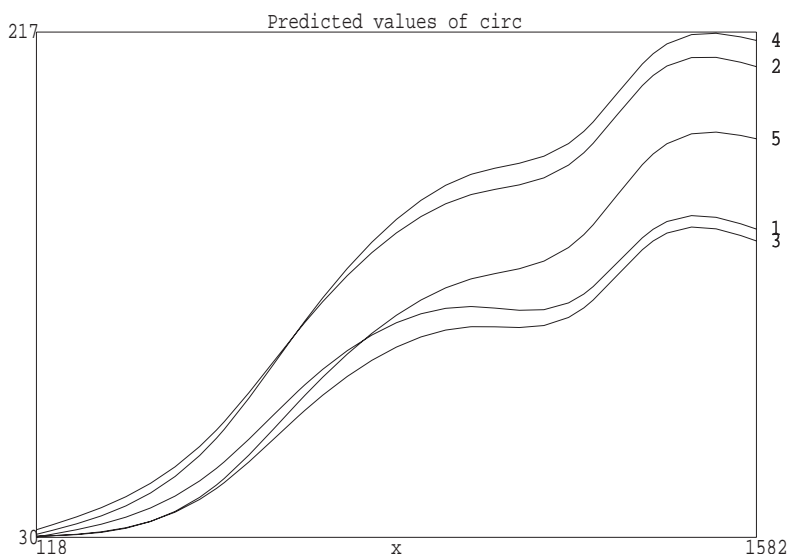


Figure 15.14 Plot of fitted cubic smoothing spline for model 1

of 107.5 ($P < 0.01$) while the `fac(x)` term goes to the boundry. Notice that the inclusion of the fixed term `season` in models 3 to 6 means that comparisons with models 1 and 2 on the basis of the log-likelihood are not valid. The spring measurements are lower than the autumn measurements so growth is slower in winter. Models 4 and 5 successively examined each term, indicating that both smoothing constants are significant ($P < 0.05$). Lastly we add the covariance parameter between the intercept and slope for each tree in model 6. This ensures that the covariance model will be translation invariant. A portion of the output file for model 6 is presented below.

```

11 LogL=-87.4291      S2=  5.6329      32 df

Source              Model  terms      Gamma      Component      Comp/SE      % C
spl(x,7)             5      5      2.17207      12.2350      1.09      0 P
spl(x,7).Tree       25     25     1.38506      7.80187      1.47      0 P
Variance             35     32     1.00000      5.63286      1.72      0 P
Tree                 UnStruct  1     5.62102      31.6624      1.26      0 U
Tree                 UnStruct  1    -0.124177E-01 -0.699474E-01 -0.85      0 U
Tree                 UnStruct  2     0.108357E-03 0.610357E-03  1.40      0 U
Covariance/Variance/Correlation Matrix UnStructured
5.621      -0.5032
-0.1242E-01 0.1084E-03

Analysis of Variance              DF      F-incr
8 mu                              1      169.87

```

5 x	1	92.78
6 Season	1	108.57

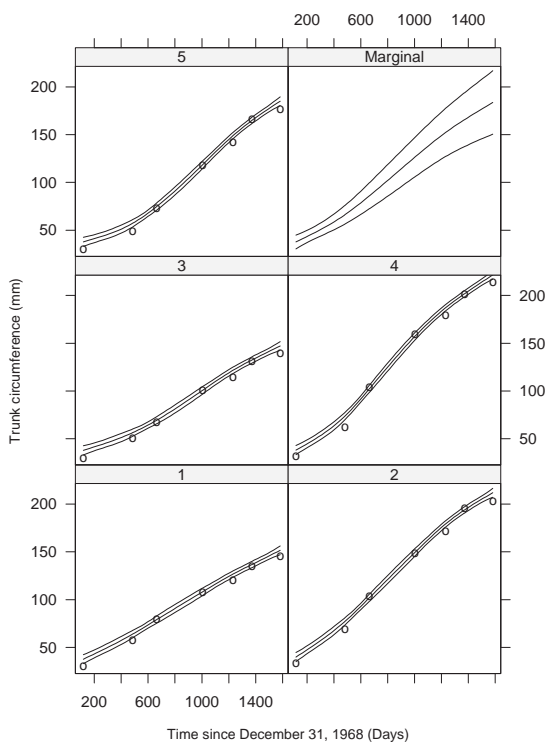


Figure 15.15: Trellis plot of trunk circumference for each tree at sample dates (adjusted for *season* effects), with fitted profiles across time and confidence intervals

Figure 15.15 presents the predicted growth over time for individual trees and a marginal prediction for trees with approximate confidence intervals ($2\pm\times$ standard error of prediction). Within this figure, the data is adjusted to remove the estimated seasonal effect. The conclusions from this analysis are quite different from those obtained by the nonlinear mixed effects analysis. The individual curves for each tree are not convincingly modelled by a logistic function. Figure 15.16 presents a plot of the residuals from the nonlinear model fitted on p340 of Pinheiro and Bates (2000). The distinct pattern in the residuals, which is the same for all trees is taken up in our analysis by the *season* term.

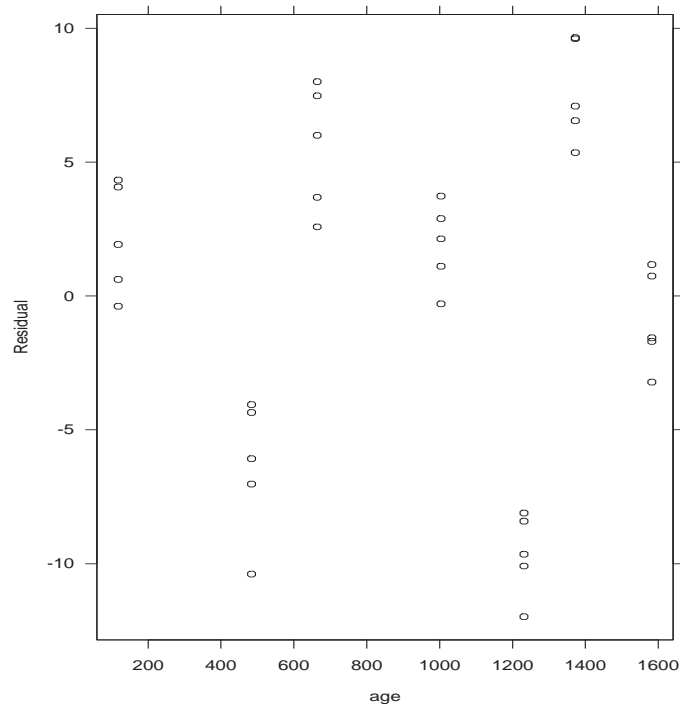


Figure 15.16 Plot of the residuals from the nonlinear model of Pinheiro and Bates

15.10 Multivariate animal genetics data - Sheep

The analysis of incomplete or unbalanced multivariate data often presents computational difficulties. These difficulties are exacerbated by either the number of random effects in the linear mixed model, the number of traits, the complexity of the variance models being fitted to the random effects or the size of the problem. In this section we illustrate two approaches to the analysis of a complex set of incomplete multivariate data.

Much of the difficulty in conducting these types of analyses in ASReml is centred on obtaining good starting values. Derivative based algorithms such as the AI algorithm can be unreliable when fitting complex variance structures unless good starting values are available. Poor starting values may result in divergence of the algorithm or slow convergence. A particular problem with fitting unstructured variance models is keeping the estimated variance matrix positive definite. These are not simple issues and in the following we present a pragmatic approach which

attempts to overcome some of these problems.

The data are taken from a large genetic study on Coopworth lambs. A total of 5 traits, namely weaning weight (**wwt**), yearling weight (**ywt**), greasy fleece weight (**gfw**), fibre diameter (**fdm**) and ultrasound fat depth at the C site (**fat**) were measured on 7043 lambs. The lambs were the progeny of 92 sires and 3561 dams, produced from 4871 litters over 49 flock-year combinations. Not all traits were measured on each group. No pedigree data was available for either sires or dams.

The aim of the analysis is to estimate heritability (h^2) of each trait and to estimate the genetic correlations between the five traits. We will present two approaches, a half-sib analysis and an analysis based on the use of an animal model, which directly defines the genetic covariance between the progeny and sires and dams.

The data fields included factors defining sire, dam and lamb (**tag**), covariates such as **age**, the age of the lamb at a set time, **brr** the birth rearing rank (1 = born single raised single, 2 = born twin raised single, 3 = born twin raised twin and 4 = other), **sex** (M, F) and **grp** a factor indicating the flock-year combination.

Half-sib analysis

In the half-sib analysis we include terms for the random effects of **sires**, **dams** and **litters**. In univariate analyses the variance component for **sires** is denoted by $\sigma_s^2 = \frac{1}{4}\sigma_A^2$ where σ_A^2 is the additive genetic variance, the variance component for **dams** is denoted by $\sigma_d^2 = \frac{1}{4}\sigma_A^2 + \sigma_m^2$ where σ_m^2 is the maternal variance component and the variance component for **litters** is denoted by σ_l^2 and represents variation attributable to the particular mating.

For a multivariate analysis these variance components for **sires**, **dams** and **litters** are, in theory replaced by unstructured matrices, one for each term. Additionally we assume the residuals for each trait may be correlated. Thus for this example we would like to fit a total of 4 unstructured variance models. For such a situation, it is sensible to commence the modelling process with a series of univariate analyses. These give starting values for the diagonals of the variance matrices, but also indicate what variance components are estimable. The ASReml job for the univariate analyses is given below

```
Multivariate Sire & Dam model
tag
sire 92 !I
dam 3561 !I
```

Table 15.14: REML estimates of a subset of the variance parameters for each trait for the genetic example, expressed as a ratio to their asymptotic s.e.

term	wwt	ywt	gfw	fdm	fat
sire	3.68	3.57	3.95	1.92	1.92
dam	6.25	4.93	2.78	0.37	0.05
litter	8.79	0.99	2.23	1.91	0.00
age.grp	2.29	1.39	0.31	1.15	1.74
sex.grp	2.90	3.43	3.70	-	1.83

Table 15.15 Wald tests of the fixed effects for each trait for the genetic example

term	wwt	ywt	gfw	fdm	fat
age	331.3	67.1	52.4	2.6	7.5
brr	554.6	73.4	14.9	0.3	13.9
sex	196.1	123.3	0.2	2.9	0.6
age.sex	10.3	1.7	1.9	-	5.0

```

grp    49
sex
brr    4
litter 4871
age    wwt  !MO ywt  !MO  # !MO recodes zeros as missing values
gfw    !MO fdm !MO fat  !MO
./wwt/coop.bin
wwt ~ mu age brr sex age.sex !r sire dam lit age.grp sex.grp !f grp

```

Tables 15.14 and 15.15 present the summary of these analyses. Fibre diameter was measured on only 2 female lambs and so interactions with `sex` were not fitted. The dam variance component was quite small for both fibre diameter and fat. The REML estimate of the variance component associated with litters was

effectively zero for fat.

Thus in the multivariate analysis we consider fitting the following models to the sire, dam and litter effects,

$$\begin{aligned}\text{var}(\mathbf{u}_s) &= \boldsymbol{\Sigma}_s \otimes \mathbf{I}_{92} \\ \text{var}(\mathbf{u}_d) &= \boldsymbol{\Sigma}_d \otimes \mathbf{I}_{3561} \\ \text{var}(\mathbf{u}_l) &= \boldsymbol{\Sigma}_l \otimes \mathbf{I}_{4891}\end{aligned}$$

where $\boldsymbol{\Sigma}_s^{5 \times 5}$, $\boldsymbol{\Sigma}_d^{3 \times 3}$ and $\boldsymbol{\Sigma}_l^{4 \times 4}$ are positive definite symmetric matrices corresponding to the between traits variance matrices for sires, dams and litters respectively. The variance matrix for dams does not involve fibre diameter and fat depth, while the variance matrix for litters does not involve fat depth. The effects in each of the above vectors are ordered levels within traits. Lastly we assume that the residual variance matrix is given by

$$\boldsymbol{\Sigma}_e \otimes \mathbf{I}_{7043}$$

Table 15.16 presents the sequence variance models fitted to each of the four random terms `sire`, `dam`, `litter` and `error` in the ASReml job presented below.

```
Multivariate Sire & Dam model
tag
sire 92 !I
dam 3561 !I
grp 49
sex
brr 4
litter 4871
age wwt !m0 ywt !m0 # !M0 identifies missing values
gfw !m0 fdm !m0 fat !m0
./wwt/coop.bin !DOPART $1 !CONTINUE !MAXIT 20
!PART 3
!EXTRA 4
!PART
wwt ywt gfw fdm fat ~ Trait Tr.age Tr.brr Tr.sex Tr.age.sex,
!r Tr.sire,
at(Tr,1).dam at(Tr,2).dam at(Tr,3).dam,
at(Tr,1).lit at(Tr,2).lit at(Tr,3).lit at(Tr,4).lit,
at(Trait,1).age.grp .0024,
at(Trait,2).age.grp .0019,
at(Trait,4).age.grp .0020,
at(Trait,5).age.grp .00026,
at(Trait,1).sex.grp .93,
at(Trait,2).sex.grp 16.0,
```

```

        at(Trait,3).sex.grp .28,
        at(Trait,5).sex.grp 1.18,
!f Tr.grp

1 2 3                #1 R structure with 2 dimensions and 3 G structures

0 0 0                #Independent across animals

Tr 0 US              #General structure across traits
15*0.                #Asreml will estimate some starting values

Tr.sire 2            #Sire effects.

!PART 1              #Initial analysis ignoring genetic correlations
Tr 0 DIAG            #Specified diagonal variance structure
0.608 1.298 0.015 0.197 0.035      #Initial sire variances

!PART 2              #Factor Analytic model
Tr 0 FA1 !GP
    0.5 0.5 -.01 -.01 0.1          #Correlation factors
0.608 1.298 0.015 0.197 0.035      #Variances

!PART 3              #Unstructured variance model
Tr 0 US
0.6199                #Lower triangle row-wise
    0.6939          1.602
    0.003219 0.007424 0.01509
-0.02532 -0.05840 -0.0002709 0.1807
    0.06013 0.1387      0.0006433 -0.005061 0.03487
!PART
sire

#Maternal structure covers the 3 model terms
#          at(Tr,1).dam at(Tr,2).dam at(Tr,3).dam

at(Tr,1).dam 2        # Maternal effects.
!PART 1
3 0 CORGH !GU         # Equivalent to Unstructured
.9
.1 .1
2.2 4.14 0.018
!PART 2
3 0 CORGH !GU
.9
.1 .1
2.2 4.14 0.018
!PART 3
3 0 US !GU
.9
.1 .1
2.2 4.14 0.018

```

```

!PART
dam

#Litter structure covers the 4 model terms at(Tr,1).lit at(Tr,2).lit
#at(Tr,3).lit at(Tr,4).lit

at(Tr,1).lit 2          # Litter effects.
!PART 1
4 0 DIAG              # Diagonal structure
3.74 0.97 0.019 0.941
!PART 2
4 0 FA1 !GP          # Factor Analytic 1
.5 .5 .01 .1
4.95 4.63 0.037 0.941
!PART 3
4 0 US                # Unstructured
5.073
 3.545      3.914
0.1274      0.08909 0.02865
0.07277 0.05090 0.001829 1.019

!PART
lit

```

Table 15.16: Variance models fitted for each part of the ASReml job in the analysis of the genetic example

term	matrix	!Part 1	!Part 2	!Part 3
sire	Σ_s	DIAG	FA1	US
dam	Σ_d	CORGH	CORGH	US
litter	Σ_l	DIAG	FA1	US
error	Σ_e	US	US	US

In !PART 1, the error variance model is taken to be unstructured, but the starting values are set to zero. This instructs ASReml to obtain starting values from the sample covariance matrix of the data. For incomplete data the matrix so obtained may not, in general be positive definite. Care should be taken when using this option for incomplete multivariate data. The command to run this file for !PART 1 is

```
asreml -nrs3 mt 1
```

The results from this analysis can be automatically used by ASReml for the next part, if the .rsv is copied prior to running the next part. That is, we add the !PART 2 coding to the job, copy mt1.rsv to mt2.rsv so that when we run !PART 2 it starts from where !PART 1 finished, and run the job using

```
asreml -cnrs3 mt 2
```

Finally, we use the !PART 3 coding to obtain the final analysis, copy mt2.rsv to mt3.rsv and run the final stage starting from the stage 2 results. Note that we are using the automatic updating associated with !CONTINUE.

A portion of the final output file is presented below.

```
NOTICE: 165 (more) singularities,
LogL=-21425.9 S2= 1.0000 35006 df
LogL=-21423.7 S2= 1.0000 35006 df
LogL=-21420.9 S2= 1.0000 35006 df
LogL=-21420.1 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df
LogL=-21419.9 S2= 1.0000 35006 df

Source      Model terms      Gamma      Component      Comp/SE      % C
at(Tr,1).age.grp 49 49 0.135358E-02 0.135358E-02 2.03 0 P
at(Tr,2).age.grp 49 46 0.101560E-02 0.101560E-02 1.24 0 P
at(Tr,4).age.grp 49 11 0.176513E-02 0.176513E-02 1.13 0 P
at(Tr,5).age.grp 49 28 0.209278E-03 0.209278E-03 1.68 0 P
at(Tr,1).sex.grp 49 49 0.919610 0.919610 2.89 0 P
at(Tr,2).sex.grp 49 46 15.3912 15.3912 3.50 0 P
at(Tr,3).sex.grp 49 46 0.279496 0.279496 3.71 0 P
at(Tr,5).sex.grp 49 28 1.44032 1.44032 1.80 0 P
.
.
.
.
.
.
Covariance/Variance/Correlation Matrix US=UnStructu
9.462 0.5691 0.2356 0.1640 0.2183
7.332 17.54 0.4241 0.2494 0.4639
0.2728 0.6686 0.1417 0.3994 0.1679
0.9625 1.994 0.2870 3.642 0.4875E-01
0.8336 2.412 0.7846E-01 0.1155 1.541
Covariance/Variance/Correlation Matrix US=UnStructu
0.5941 0.7044 0.2966 0.2028 0.2703
0.6745 1.544 0.1364E-01-0.1227 0.5726
0.2800E-01 0.2076E-02 0.1500E-01 0.1114 -0.4818E-02
```

```

0.6224E-01-0.6072E-01 0.5433E-02 0.1586    -0.6334
0.3789E-01 0.1294    -0.1073E-03-0.4588E-01 0.3308E-01
Covariance/Variance/Correlation Matrix US=UnStructu
 2.161      1.010      0.7663
 2.196      2.186      0.8301
0.1577      0.1718      0.1959E-01
Covariance/Variance/Correlation Matrix US=UnStructu
 3.547      0.5065     -0.1099     -0.4097E-01
 1.555      2.657      0.1740     -0.5150
-0.2787E-01 0.3822E-01 0.1815E-01-0.3281
-0.7313E-01-0.7957     -0.4190E-01 0.8984

Analysis of Variance      DF      F-incr
15 Tr.age                5        98.95
16 Tr.brr                15       116.72
17 Tr.sex                5        59.78
19 Tr.age.sex            4         4.90

```

The REML estimates of all the variance matrices are positive definite. Heritabilities for each trait can be calculated using the `.pin` file facility of ASReml. The heritability is given by

$$h^2 = \frac{\sigma_A^2}{\sigma_P^2}$$

where σ_P^2 is the phenotypic variance and is given by

$$\sigma_P^2 = \sigma_s^2 + \sigma_d^2 + \sigma_l^2 + \sigma_e^2$$

recalling that

$$\begin{aligned}\sigma_s^2 &= \frac{1}{4}\sigma_A^2 \\ \sigma_d^2 &= \frac{1}{4}\sigma_A^2 + \sigma_m^2\end{aligned}$$

In the half-sib analysis we only use the estimate of additive genetic variance from the sire variance component. The ASReml `.pin` file is presented below along with the output from the following command

```
asreml -p mt3
```

```

F phenWYG 9:14 + 24:29 + 39:44 + 45:50    # defines 55:60
F phenD   15:18 + 30:33 +                51:54    # defines 61:64
F phenF   19:23 + 34:38                    # defines 65:69

```



```

F Direct 24:38 * 4. # defines 70:84
F Maternal 39:44 - 24:29 # defines 85:90
H WTh2 70 55
H YTh2 72 57
H GFWh2 75 60
H FDMh2 79 64
H FATH2 84 69
R GenCor 24:38
R MatCor 85:90

55 phenWYG 9 15.76 0.3130
56 phenWYG 10 11.76 0.3749
57 phenWYG 11 23.92 0.6313
. . .
70 Direct 24 2.376 0.6458
71 Direct 25 2.698 0.8487
72 Direct 26 6.174 1.585
73 Direct 27 0.1120 0.7330E-01
. . .
85 Maternal 39 1.567 0.3788
86 Maternal 40 1.521 0.4368
87 Maternal 41 0.6419 0.7797
. . .
WTh2 = Direct 2 70/phenWYG 55= 0.1507 0.0396
YTh2 = Direct 2 72/phenWYG 57= 0.2581 0.0624
GFWh2 = Direct 2 75/phenWYG 60= 0.3084 0.0716
FDMh2 = Direct 3 79/phenD 18 64= 0.1350 0.0717
FATH2 = Direct 3 84/phenF 23 69= 0.0841 0.0402
GenCor 2 1 = Tr.si 25/SQR[Tr.si 24*Tr.si 26]= 0.7044 0.1025
GenCor 3 1 = Tr.si 27/SQR[Tr.si 24*Tr.si 29]= 0.2966 0.1720
GenCor 3 2 = Tr.si 28/SQR[Tr.si 26*Tr.si 29]= 0.0136 0.1810
GenCor 4 1 = Tr.si 30/SQR[Tr.si 24*Tr.si 33]= 0.2028 0.3513
GenCor 4 2 = Tr.si 31/SQR[Tr.si 26*Tr.si 33]= -0.1227 0.3247
GenCor 4 3 = Tr.si 32/SQR[Tr.si 29*Tr.si 33]= 0.1115 0.3868
GenCor 5 1 = Tr.si 34/SQR[Tr.si 24*Tr.si 38]= 0.2703 0.2724
GenCor 5 2 = Tr.si 35/SQR[Tr.si 26*Tr.si 38]= 0.5726 0.2022
GenCor 5 3 = Tr.si 36/SQR[Tr.si 29*Tr.si 38]= -0.0048 0.2653
GenCor 5 4 = Tr.si 37/SQR[Tr.si 33*Tr.si 38]= -0.6333 0.3775
MatCor 2 1 = Mater 86/SQR[Mater 85*Mater 87]= 1.5168 0.7131
MatCor 3 1 = Mater 88/SQR[Mater 85*Mater 90]= 1.5285 1.1561
MatCor 3 2 = Mater 89/SQR[Mater 87*Mater 90]= 3.1251 2.7985

```

Animal model

In this section we will illustrate the use of `.ped` files to define the genetic pedigree structure between animals. This is an alternate method of estimating additive genetic variance for these data. The variance matrix of the animals (sires, dams

and lambs) for which we only have data on lambs is given by

$$\text{var}(\mathbf{u}_A) = \Sigma_A \otimes \mathbf{A}^{-1}$$

where \mathbf{A}^{-1} is the inverse of the genetic relationship matrix. There are a total of $10696 = 92 + 3561 + 7043$ animals in the pedigree. The ASReml input file is presented below. Note that this model is not equivalent to the sire/dam/litter model with respect to the animal/litter components for gfw, fd and fat.

```
Multivariate Animal model
tag !P
sire
dam !P
grp 49
sex
brr 4
litter 4871
age wwt !m0 ywt !m0 # !M0 identifies missing values
gfw !m0 fdm !m0 fat !m0
./wwt/coop.ped
./wwt/coop.bin !DOPART $1 !CONTINUE !MAXIT 20 !RECODE !STEP 0.01
# $1 allows selection of PART as a command line argument
!PART 3
!EXTRA 4 # Force 4 more iterations after convergence criterion met
!PART
wwt ywt gfw fdm fat ~ Trait Tr.age Tr.brr Tr.sex Tr.age.sex,
!r Tr.tag ,
at(Tr,1).dam, at(Tr,2).dam, -at(Tr,3).dam .003,
at(Tr,1).lit, at(Tr,2).lit, at(Tr,3).lit, at(Tr,4).lit,
at(Trait,1).age.grp .0024,
at(Trait,2).age.grp .0019,
at(Trait,4).age.grp .0020,
at(Trait,5).age.grp .00026,
at(Trait,1).sex.grp .93,
at(Trait,2).sex.grp 16.0,
at(Trait,3).sex.grp .28,
at(Trait,5).sex.grp 1.18,
!f Tr.grp

1 2 3 # One multivariate R structure, 3 G structures

0 0 0 # No structure across lamb records
# First zero lets ASReml count the number of records
Tr 0 US #General structure across traits
7.66
5.33 13
.18 .66 .10
.78 2.1 .27 3.2
.73 2.02 .08 .20 1.44
```

```

Tr.tag 2                      # Direct animal effects.

!PART 2
Tr 0 FA1 !GP
    0.5 0.5 -.01 -.01 0.1
2.4 5.2 0.06 .8 .14

!PART 3
Tr 0 US
2.4800
    2.8 6.4
0.0128 0.03 0.06
-.1 -.22 -.0011 0.72
    0.24 0.55      0.0026 -0.0202 0.14
!PART
tag

at(Tr,1).dam 2                # Maternal effects.
!PART 2
2 0 CORGH !GFU
.99
1.6 2.54
!PART 3
2 0 US !GU
1.1 .58 .31
!PART
dam

at(Tr,1).lit 2                # Litter effects.
!PART 2
4 0 FA1 !GP                    # Factor Analytic
.5 .5 .01 .1 .01
4.95 4.63 0.037 0.941 0.102
!PART 3
4 0 US                          # Unstructured
5.073
    3.545      3.914
    0.1274      0.08909 0.02865
    0.07277 0.05090 0.001829 1.019

!PART
lit

```

The term `Tr.tag` now replaces the `sire` (and part of `dam`) terms in the half-sib analysis. This analysis uses information from both sires and dams to estimate additive genetic variance. The dam variance component in this analysis only estimates the maternal variance component. It is only significant for the weaning and yearling weights. The litter variation remains unchanged. The ASReml input

file again consists of several parts, which progressively build up to fitting unstructured variance models to `Tr.tag`, `Tr.dam`, `Tr.litter` and error. A portion of the output file is presented below.

```

dam !P
age      wwt  !m0 ywt  !m0
gfw      !m0 fdm !m0 fat  !m0
A-inverse retrieved from ainverse.bin
PEDIGREE [../wvt/coop.ped ] has 10696 identities, 29474 Non zero elements
QUALIFIERS: !DOPART 3 !CONTINUE !MAXIT 20 !RECODE !STEP 0.01
Reading ../wvt/coop.bin as BINARY SINGLE PRECISION

Model term      Size Type      COL  Minimum      Mean      Maximum      #zero #miss
1 tag           10696 Direct      1    3.000        5380.     0.1070E+05    0     0
2 sire           1 Covariat      2    1.000        48.06     92.00         0     0
3 dam           10696 Direct      3    1.000        5197.     0.1070E+05    0     0
.
.
.
Forming 95033 equations: 40 dense
Initial updates will be shrunk by factor 0.010
NOTICE: 165 (more) singularities,
LogL=-21432.6 S2= 1.0000 35006 df
LogL=-21432.3 S2= 1.0000 35006 df
LogL=-21429.6 S2= 1.0000 35006 df
LogL=-21423.2 S2= 1.0000 35006 df
LogL=-21420.1 S2= 1.0000 35006 df
LogL=-21420.1 S2= 1.0000 35006 df
LogL=-21417.6 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
LogL=-21417.2 S2= 1.0000 35006 df
.
.
.
Source      Model terms      Gamma      Component      Comp/SE      % C
at(Tr,1).age.grp 49 49 0.132682E-02 0.132682E-02 2.02 0 P
at(Tr,2).age.grp 49 46 0.908219E-03 0.908219E-03 1.15 0 P
at(Tr,4).age.grp 49 11 0.175614E-02 0.175614E-02 1.13 0 P
at(Tr,5).age.grp 49 28 0.223617E-03 0.223617E-03 1.73 0 P
at(Tr,1).sex.grp 49 49 0.902586 0.902586 2.88 0 P
at(Tr,2).sex.grp 49 46 15.3623 15.3623 3.50 0 P
at(Tr,3).sex.grp 49 46 0.280673 0.280673 3.71 0 P
at(Tr,5).sex.grp 49 28 1.42136 1.42136 1.80 0 P
.
.
.
Covariance/Variance/Correlation Matrix US=UnStructu
7.476 0.4918 0.1339 0.1875 0.1333
4.768 12.57 0.4381 0.3425 0.3938

```

```

0.1189    0.5049    0.1056    0.4864    0.1298
0.9377    2.221     0.2891    3.345     0.1171
0.4208    1.612     0.4869E-01 0.2473    1.333
Covariance/Variance/Correlation Matrix US=UnStructu
3.898    0.8164    0.5763    0.3898E-01 0.6148
4.877    9.154     0.3689    -0.1849    0.7217
0.3029    0.2971    0.7085E-01-0.2416E-01 0.3041
0.6019E-01-0.4375    -0.5030E-02 0.6117    -0.4672
0.6154    1.107     0.4104E-01-0.1853    0.2570
Covariance/Variance/Correlation Matrix US=UnStructu
0.9988    0.7024
0.5881    -0.7018
Covariance/Variance/Correlation Matrix US=UnStructu
3.714    0.5511    0.1635    -0.6157E-01
2.019    3.614     0.5176    -0.4380
0.4506E-01 0.1407    0.2045E-01-0.3338
-0.1021    -0.7166    -0.4108E-01 0.7407

Analysis of Variance      DF      F-incr
15 Tr.age                5       99.16
16 Tr.brr                15      116.52
17 Tr.sex                5       59.94
19 Tr.age.sex            4       5.10

```

There is no guarantee that variance component matrices will be positive definite and this example highlights this issue. We used the !GU qualifier on the maternal component to obtain the matrix

$$\begin{bmatrix} 0.9988 & 0.5881 \\ 0.5881 & -0.7018 \end{bmatrix}.$$

ASReml reports the correlation as 0.7024 which it obtains by ignoring the sign in -0.7018. This is the maternal component for ywt. Since it is entirely reasonable to expect maternal influences on growth to have dissipated at 12 months of age, it would be reasonable to refit the model omitting `at(Tr,2).dam` and changing the dimension of the G structure.

Bibliography

- Cox, D. R. and Hinkley, D. V. (1974). *Theoretical Statistics*, London: Chapman and Hall.
- Cox, D. R. and Snell, E. J. (1981). *Applied Statistics; Principles and Examples*, London: Chapman and Hall.
- Cressie, N. A. C. (1991). *Statistics for spatial data*, New York: John Wiley and Sons, Inc.
- Cullis, B. R. and Gleeson, A. C. (1991). Spatial analysis of field experiments - an extension to two dimensions, *Biometrics* **47**: 1449–1460.
- Cullis, B. R., Gleeson, A. C., Lill, W. J., Fisher, J. A. and Read, B. J. (1989). A new procedure for the analysis of early generation variety trials, *Applied Statistics* **38**: 361–375.
- Cullis, B. R., Gogel, B. J., Verbyla, A. P. and Thompson, R. (1998). Spatial analysis of multi-environment early generation trials, *Biometrics* **54**: 1–18.
- Dempster, A. P., Selwyn, M. R., Patel, C. M. and Roth, A. J. (1984). Statistical and computational aspects of mixed model analysis, *Applied Statistics* **33**: 203–214.
- Draper, N. R. and Smith, H. (1998). *Applied Regression Analysis*, John Wiley and Sons, New York, 3rd Edition.
- Gilmour, A. R., Cullis, B. R. and Verbyla, A. P. (1997). Accounting for natural and extraneous variation in the analysis of field experiments, *Journal of Agricultural, Biological, and Environmental Statistics* **2**: 269–273.
- Gilmour, A. R., Cullis, B. R., Welham, S. J., Gogel, B. J. and Thompson, R. (2002). An efficient computing strategy for prediction in mixed linear models, *Computational Statistics and Data Analysis*.
- Gilmour, A. R., Thompson, R. and Cullis, B. R. (1995). AI, an efficient algorithm for REML estimation in linear mixed models, *Biometrics* **51**: 1440–1450.

- Gleeson, A. C. and Cullis, B. R. (1987). Residual maximum likelihood (REML) estimation of a neighbour model for field experiments, *Biometrics* **43**: 277–288.
- Gogel, B. J. (1997). *Spatial analysis of multi-environment variety trials*, PhD thesis, Department of Statistics, University of Adelaide.
- Green, P. J. and Silverman, B. W. (1994). *Nonparametric regression and generalized linear models*, London: Chapman and Hall.
- Harvey, W. R. (1977). *Users' guide to LSML76*, The Ohio State University, Columbus.
- Harville, D. A. (1997). *Matrix algebra from a statisticians perspective*, Springer-Verlag, New York.
- Kenward, M. G. and Roger, J. H. (1997). The precision of fixed effects estimates from restricted maximum likelihood, *Biometrics* **53**: 983–997.
- Lane, P. W. and Nelder, J. A. (1982). Analysis of covariance and standardisation as instances of prediction, *Biometrics* **38**: 613–621.
- Patterson, H. D. and Nabugoomu, F. (1992). REML and the analysis of series of crop variety trials, *Proceedings from the 25th International Biometric Conference*, pp. 77–93.
- Patterson, H. D. and Thompson, R. (1971). Recovery of interblock information when block sizes are unequal, *Biometrika* **58**: 545–54.
- Piepho, H.-P., Denis, J.-B. and van Eeuwijk, F. A. (1998). Mixed biadditive models, *Proceedings of the 28th International Biometrics Conference*.
- Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-PLUS*, Berlin: Springer-Verlag.
- Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects, *Statistical Science* **6**: 15–51.
- Searle, S. R. (1971). *Linear Models*, New York: John Wiley and Sons, Inc.
- Searle, S. R. (1982). *Matrix algebra useful for statistics*, New York: John Wiley and Sons, Inc.
- Searle, S. R., Casella, G. and McCulloch, C. E. (1992). *Variance Components*, New York: John Wiley and Sons, Inc.

- Self, S. C. and Liang, K. Y. (1987). Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under non-standard conditions, *Journal of the American Statistical Society* **82**: 605–610.
- Smith, A. B., Cullis, B. R., Gilmour, A. R. and Thompson, R. (1998). Multiplicative models for interaction in spatial mixed model analyses of multi-environment trial data, *Proceedings of the 28th International Biometrics Conference*.
- Smith, A., Cullis, B. R. and Thompson, R. (2001). Analysing variety by environment data using multiplicative mixed models and adjustments for spatial field trend, *Biometrics* **57**: 1138–1147.
- Stevens, M. M., Fox, K. M., Warren, G. N., Cullis, B. R., Coombes, N. E. and Lewin, L. G. (1999). An image analysis technique for assessing resistance in rice cultivars to root-feeding chironomid midge larvae (diptera: Chironomidae), *Field Crops Research* **66**: 25–26.
- Stroup, W. W., Baenziger, P. S. and Mulitze, D. K. (1994). Removing spatial variation from wheat yield trials: a comparison of methods, *Crop Science* **86**: 62–66.
- Thompson, R., Cullis, B. R., Smith, A. B. and Gilmour, A. R. (2003). A sparse implementation of the average information algorithm for factor analytic and reduced rank variance models. (submitted).
- Verbyla, A. P., Cullis, B. R., Kenward, M. G. and Welham, S. J. (1999). The analysis of designed experiments and longitudinal data using smoothing splines, *Journal of the Royal Statistical Society, Series C* **48**: 269–311.
- White, I. H., Thompson, R. and Brotherstone, S. (1999). Genetic and environmental smoothing of lactation curves with cubic splines, *Journal of Dairy Science* **82**: 632–638.
- Wilkinson, G. N. and Rogers, C. E. (1973). Symbolic description of factorial models for analysis of variance, *Applied Statistics* **22**: 392–399.
- Wolfinger, R. D. (1996). Heterogeneous variance-covariance structures for repeated measures, *Journal of Agricultural, Biological, and Environmental Statistics* **1**: 362–389.
- Yates, F. (1935). Complex experiments, *Journal of the Royal Statistical Society, Series B* **2**: 181–247.

Index

- advanced processing arguments, 147
- AI algorithm, 12
- AIC, 17
- Akaike Information Criteria, 17
- aliasing, 74
- animal breeding data, 2
- arguments, 4
- ASReml symbols
 - \sim , 62
 - $*$, 35
 - $.$, 35
 - $\#$, 35
 - $\$$, 35
 - $!\{$, 64
 - $!\}$, 64
 - $+$, 64
 - $,,$ 64
 - $-$, 64
 - $/$, 64
 - $:$, 64
- autoregressive, 83
- Average Information, 2
- balanced repeated measures, 204
- Bayesian Information Criteria, 17
- BIC, 17
- binary files, 36
- BLUE, 14
- BLUP, 14
- case, 63
- combining variance models, 15
- command file, 23
 - genetic analysis, 118
 - multivariate, 111
- command line options, 142
- commonly used functions, 64
- conditional distribution, 11
- conditional factors, 68
- constraining variance parameters, 106
- constraints
 - on variance parameters, 87
- Convergence criterion, 49
- correlated random effects, 14
- correlation, 137
 - between traits, 111
 - model, 9
- covariance model, 10
 - isotropic, 9
- covariates, 34, 47, 73
- cubic splines, 71
- data field syntax, 39
- data file, 22, 33, 34
 - binary format, 36
 - fixed format, 36
 - free format, 34
 - line, 47
- data file line, 25
 - qualifiers, 48
 - syntax, 47
- debug options, 143
- dense, 74
- design factors, 73
- diagnostics, 17
- diallal analysis, 69
- direct product, 6, 8, 78
- direct sum, 8
- discussion list, 3
- distribution
 - conditional, 11
 - marginal, 11

- Ecode, 32
- EM update, 103
- equations
 - mixed model, 13
- error variance
 - heterogeneity, 8
- errors, 169
- expected information matrix, 12

- F-statistic, 18
- factors, 34
- file
 - GIV, 123
 - pedigree, 119
- Fisher-scoring algorithm, 12
- fixed effects, 6
- Fixed format files, 56
- fixed terms, 62, 66
 - multivariate, 112
 - primary, 66
 - sparse, 67
- free format, 34
- functions of variance components, 31, 134
 - correlation, 137
 - heritability, 136
 - linear combinations, 135
 - syntax, 135

- G structure, 78
 - definition lines, 87, 92
 - header, 92
 - more than one term, 105
- genetic
 - data, 2
 - groups, 121
 - links, 118
 - models, 118
 - qualifiers, 118
 - relationships, 119
- GIV, 123

- graphics options, 143
- half-sib analysis, 246
- help, 3
- heritability, 136, 165
- heterogeneity
 - error variance, 8
- hypothesis testing, 18

- identifiable, 15
- IID, 6
- information matrix
 - expected, 12
 - observed, 12
- initial values, 91
- input file extension
 - .BIN, 36
 - .DBL, 36
 - .bin, 34, 36
 - .csv, 35
 - .dbl, 34, 36
 - .pin, 135
- interactions, 68
- isotropic
 - covariance model, 9

- job control
 - options, 144
 - qualifiers, 49

- key output files, 151

- likelihood
 - log residual, 11
 - residual, 11
- Log file, 146
- longitudinal data, 2
 - balanced example, 237

- marginal distribution, 11
- measurement error, 84
- memory options, 144

- menu options, 145
- MET, 8
- meta analysis, 2, 8
- missing values, 35, 71, 73, 155
 - NA, 35
 - in explanatory variables, 73
 - in response, 73
- mixed
 - effects, 6
 - model, 6
- mixed model, 6
 - equations, 13
 - multivariate, 112
 - specifying, 25
- model
 - animal, 118, 253
 - correlation, 9
 - covariance, 10
 - formulae, 62
 - random regression, 10
 - sire, 118
- model building, 109
- moving average, 70
- multi-environment trial, 2, 8
- multivariate analysis, 111, 231
 - example, 245
 - half-sib analysis, 246
- Nebraska Intrastate Nursery, 20
- non singular matrices, 78
- nonidentifiable, 15
- objective function, 13
- observed information matrix, 12
- operators, 64
- options
 - command line, 142
- ordering of terms, 74
- orthogonal polynomials, 71
- outliers, 166
- output
 - files, 27
 - multivariate analysis, 114
 - objects, 165
- output file extension
 - .asl, 150
 - .asp, 150
 - .asr, 27, 150, 151
 - .dbr, 150
 - .dpr, 150, 162
 - .pvc, 150
 - .pvs, 150, 162
 - .res, 150, 156
 - .rsv, 150, 162
 - .sln, 29, 150, 154
 - .spr, 150
 - .tab, 150, 164
 - .veo, 150
 - .vrb, 159
 - .vvp, 150, 161
 - .yht, 30, 150, 155
- overspecified, 15
- own models, 98
- OWN variance structure, 97
 - !F2, 98
 - !T, 98
- parameter
 - scale, 6
 - variance, 6
- PC environment, 141
- pedigree, 118
 - file, 119
- post-processing
 - variance components, 146
- power, 95, 100
- predicted values, 31
- prediction, 26, 126
 - qualifiers, 131
- predictions
 - estimable, 32
- prior mean, 14

product
 direct, 8

qualifier
 !<, 43
 !<=, 43
 !<>, 43
 !==, 43
 !>, 43
 !>=, 43
 !E, 43
 !* , 43
 !+ , 43
 !- , 43
 !/ , 43
 !=s, 103
 != , 43
 !ABS, 43
 !ADJUST, 55
 !ALPHA, 121
 !ARCSIN , 43
 !ASMV, 50
 !ASUV, 50
 !A, 40
 !BLUP, 55
 !CINV, 59
 !COLFAC, 50
 !CONTINUE, 51, 109
 !COS, 43
 !CSV, 48
 !DATAFILE, 55
 !DENSE, 55
 !DF, 55
 !DISPLAY, 51
 !DOPART, 52
 !D, 43
 !EXP, 43
 !EXTRA, 59
 !FACPOINTS, 59
 !FILTER, 48
 !FORMAT, 56
 !GF, 103
 !GIV, 121
 !GP, 103
 !GROUPS, 122
 !GU, 103
 !GZ, 103
 !G, 40, 56, 58
 !I, 40
 !JOIN, 57, 58
 !Jddm, 44
 !Jmmd, 44
 !Jyyd, 44
 !KNOTS, 59
 !L, 40
 !MAKE, 122
 !MAXIT, 49
 !MAX, 44
 !MGS, 122
 !MIN, 44
 !MOD, 44
 !MVREMOVE, 52
 !M, 44
 !NA, 44
 !NOCHECK, 59
 !NOREORDER, 59
 !NOSCRATCH, 59
 !POLPOINTS, 59
 !PPOINTS, 60
 !PRINT, 57
 !PVAL, 52
 !P, 40
 !READ, 57
 !RECODE, 57
 !REPEAT, 122
 !REPORT, 57
 !RESIDUALS, 57
 !ROWFAC, 50, 53
 !S2==1, 104
 !S2==value, 104
 !S2=, 104

- !SAVE, 58
- !SCALE, 60
- !SECTION, 53
- !SELECT, 48
- !SEQ, 45
- !SET, 44
- !SIN, 43
- !SKIP, 48, 122
- !SLOW, 60
- !SPLINE, 54
- !STEP, 54
- !SUB, 45
- !VCC, 58
- !V, 45
- !X, 58
- !Y, 58
- qualifiers
 - data file line, 48
 - genetic, 118
 - job control, 49
 - variance model, 103
- R structure, 78
 - definition, 90
 - definition lines, 87
- random
 - effects, 6
 - correlated, 14
 - regressions
 - model , 10
 - terms
 - multivariate, 112
- random regressions, 105
- random terms, 62, 67
- RCB, 24
 - analysis, 79
 - design, 20
- reading the data, 24, 39
- REML, i, 2, 10, 16
- REMLRT, 16
- repeated measures, 2, 204
- reserved words, 64
 - AEXP, 101
 - AGAU, 101
 - AINV, 102
 - ANTE[1], 101
 - AR2, 99
 - ARMA, 99
 - AR[1], 99
 - CHOL[1], 101
 - CORB, 100
 - CORGB, 100
 - CORGH, 100
 - CORU, 99
 - C, 142
 - DIAG, 101
 - D, 142
 - EXP, 100
 - E, 142
 - FACV[1], 102
 - FA[1], 101
 - F, 142
 - GAU, 100
 - GIV, 102
 - Gg, 142
 - IDH, 101
 - ID, 99
 - IEUC, 100
 - IEXP, 100
 - IGAU, 100
 - I, 142
 - L, 142
 - MA2, 99
 - MA[1], 99
 - M, 142
 - N, 142
 - OWN, 101
 - P, 142
 - R, 142
 - SAR, 99
 - Ss, 142

- Trait, 64, 72
- US, 101
- XFA[1], 102
- $a(t, r)$, 69
- $\text{and}(t, r)$, 64, 69
- $\text{at}(f, n)$, 64, 69
- $\cos(v, r)$, 65, 69
- $\text{fac}(v, y)$, 64, 69
- $\text{fac}(v)$, 64, 69
- $g(f, n)$, 69
- $\text{giv}(f, n)$, 65, 69
- $i(f)$, 70
- $\text{ide}(f)$, 65, 70
- $\text{inv}(v, r)$, 65, 70
- $l(f)$, 70
- $\text{leg}(v, n)$, 65, 70
- $\text{lin}(f)$, 64, 70
- $\log(v, r)$, 65, 70
- $\text{ma1}(f)$, 65, 70
- ma1 , 65, 70
- μ , 64, 70
- mv , 64, 71
- $p(v, n)$, 71
- $\text{pol}(v, n)$, 65, 71
- $s(v[, k])$, 71
- $\sin(v, r)$, 65, 71
- $\text{spl}(v[, k])$, 64, 71
- $\text{sqrt}(v, r)$, 65, 71
- $\text{uni}(f, k)$, 72
- $\text{uni}(f, n)$, 65
- $\text{uni}(f)$, 65
- units, 64, 72
- $\text{xfa}(f, k)$, 65, 72
- residual
 - error, 6
 - likelihood, 11
- response, 62
- running the job, 27
- scale parameter, 6
- score, 12
- section, 8
- separability, 9
- separable, 83
- singularities, 74
- sparse, 74
 - sparse fixed, 62
- spatial
 - analysis, 213
 - data, 2
 - model, 82
- specifying the data, 39
- split plot design, 194
- tabulation, 26
 - syntax, 128
- testing of terms, 76
- tests of hypotheses, 18
- title line, 24, 39
- trait, 34, 111
- transformation, 41
 - syntax, 41
- typographic conventions, 4
- unbalanced
 - data, 202
 - nested design, 198
- UNIX, 141
- unreplicated trial, 219
- variance
 - parameter, 6
- variance components
 - functions of, 134
- variance header line, 87, 89
- variance model
 - combining, 15, 104
 - description, 92
 - forming from correlation models, 93
 - qualifiers, 103
 - specification, 78

- specifying, 79
- variance parameters, 10
 - constraining, 87, 106
 - between structures , 107
 - within a model , 106
- variance structures, 26, 87
 - multivariate, 113
- Wald test, 18
- weight, 62, 72
- weights, 34