

ASReml-R :: CHEAT SHEET

Getting Started

ASReml-R is designed to fit general Linear Mixed Models to data sets using complex variance models.

Installation instructions:

asreml.kb.vsnl.co.uk/knowledge-base/asreml-r-installation-guide/

For additional resources:

asreml.kb.vsnl.co.uk/article-categories/asreml-r-resources/

Reading Data

Data for analysis using ASReml-R are generally contained in a text file or a spreadsheet and are read into a **data frame** using the appropriate R functions.

```
nin89 <- read.table(file = "/path/nin89.csv",
  header = TRUE, sep = ",")
```

Variety	Id	pid	raw	Rep	nloc	yield	lat	long	Row	Column
16 LANCER	1	1101	585	1	4	29.25	4.3	19.2	16	1
17 BRULE	2	1102	631	1	4	31.55	4.3	20.4	17	1
18 REDLAND	3	1103	701	1	4	35.05	4.3	21.6	18	1
19 CODY	4	1104	602	1	4	30.1	4.3	22.8	19	1
20 ARAPAHOE	5	1105	661	1	4	33.05	4.3	24	20	1
21 NE83404	6	1106	605	1	4	30.25	4.3	25.2	21	1

Character fields and numeric fields are often not treated as factors, and you must manually convert them into factors as required.

```
nin89$Rep <- as.factor(nin89$Rep)
nin89$Row <- as.factor(nin89$Row)
nin89$Column <- as.factor(nin89$Column)
```

The function `asreml.read.table()` reads data from a text file and automatically converts variates whose names begin with a capital letter in the header line into factors.

```
nin89 <- asreml.read.table(file = "/path/nin89.csv",
  header = TRUE, sep = ",")
```

Fitting a Model

ASReml-R uses the Wilkinson and Rogers (1973) notation. The model term A:B denotes an interaction or nested effect depending on which other terms are previously included in the model.

```
asreml(
  fixed = y ~ Site,
  random = ~ Genotype + Site:Genotype,
  data = my_data)
```

Use + to add model terms

":" denotes interaction

```
asreml(
  fixed = y ~ Site,
  random = ~ Site:Genotype,
  data = my_data)
```

":" denotes nested effect

An example of the `asreml()` call for a simple randomized complete block (RCB) design using `nin89` dataframe.

```
nin89.asr <- asreml(
  fixed = yield ~ Variety,
  random = ~ idv(Rep, init = 10),
  residual = ~ idv(units, init = 50),
  na.action = na.method(x = "include"),
  data = nin89)
```

Set initial values (optional)

The **fixed** terms (required) is specified as a formula with the numerical response (**yield**) on the left of a `~` operator and the fixed factor (**Variety**) on the right.

In **random** and **residual** the factors (**Rep** and **units**) are positioned after the `~` operator; **units** is an internal experimental unit factor; and `idv()` is the homogeneous identity **variance structure**.

Missing values are allowed in design factors when **x = "include"**. And response records with missing values in the response are included by default with **y = "include"**.

Fixed Terms

The formula here specifies: response, fixed factors, interactions and covariates.

```
asreml(
  fixed = yield ~ Variety, ...)
```

intercept plus Variety

```
asreml(
  fixed = yield ~ -1 + Variety, ...)
```

-1 removes the overall mean

```
asreml(
  fixed = yield ~ 1, ...)
```

intercept-only

If the response term specified is several columns, then a multivariate/multitrait analysis is performed.

```
asreml(
  fixed = cbind(y1,..., yk) ~ trait + x + trait:x, ...)
```

multivariate general mean

Important: **trait** and **units** are reserved words

Random and Residual terms

Each random term needs to be associated to a variance model, `idv()` is used by default. The examples below use autoregressive matrices for both rows and columns.

```
sp.asr <- asreml(
  fixed = yield ~ Variety,
  random = ~ar1v(Column):ar1(Row),
  residual = ~idv(units),
  data = nin89)
```

ar1v() is a variance model

The operator ":" applies the direct product structure

ar1() is a correlation model

In the previous example, the spatial model is defined as a **G** structure, and in the following example, it is defined as an **R** structure.

```
sp.asr <- asreml(
  fixed = yield ~ Variety,
  random = ~idv(Rep),
  residual = ~ar1v(Column):ar1(Row),
  G.param = iv,
  data = nin89)
```

data must be sorted as rows within columns

G.param and R.param to set initial values, if needed

Conditional Factors

The conditional factor `at()` is used when another factor depends on specific levels on this factor.

```
asreml(...,
  random = ~ at(Site, 1):Block,
  residual = ~ dsum(~ idv(units) | Site), ...)
```

equivalent to diag(Site):Block

`dsum()` is available only for the residual formula and specifies a direct sum, one for each level of the conditioning factor.

Variance Models

ASReml-R has a wide range of variance models that can be used to specify the variance matrix of terms in the random and residual formulae. Some of them are shown below:

idv:	idh or diag:	corv: uniform
identity	diagonal	homogeneous
σ^2	σ^2	σ^2

corh: uniform heterogeneous autocorrelation



ar1: 1st order autocorrelation



corgh or us: unstructured



The class of general variance models also includes **cor**, **corb**, **corg**, **chol**, **cholc**, **ante** and the factor analytic structure (**fa**, **sfa**, **facv**, **rr**).

Adding a Genetic Correlation Matrix

```
harvey.ai <- ainverse(harvey.ped)
adg0.asr <- asreml(
  fixed = y3 ~ Line,
  random = ~vm(Calf, harvey.ai),
  residual = ~units,
  data = harvey)
```

reference factor

Ainv matrix

variance model function

The function `ainverse()` uses the method of Meuwissen and Luo (1992) to compute the inverse of the numerator relationship matrix (NRM) directly from pedigree.

ASReml-R :: CHEAT SHEET

ASReml Options

With no arguments, **asreml.options()** returns a list of settings that can be altered. The full list can be displayed by typing **help(asreml.options)**

```
asreml.options(  
  workspace = "512mb", maxit = 20,  
  gammaPar = FALSE, ai.sing = TRUE)
```

The settings are maintained per R session.

Weights

Weighted analyses are achieved by using the **weights** argument within function **asreml()**

```
asreml(..., variate in the data frame  
  weights = wt,  
  family = asr_gaussian(dispersion = 1.0), ...)
```

GLMM

The inclusion of argument **family** allows for specifying other probability distributions, fitting what is referred to as a Generalized Linear Mixed Model (GLMM).

```
binnor.asr <- asreml(  
  fixed = score4 ~ Sex + Grp,  
  random = ~Sire,  
  family = asr_binomial(link = "logit",  
    dispersion = 1, total = NULL),  
  data = binnor)
```

The options are Gaussian, Gamma, Poisson, Binomial and Negative Binomial.

Update

The **update()** function can be used if the model requires further interactions or modifications.

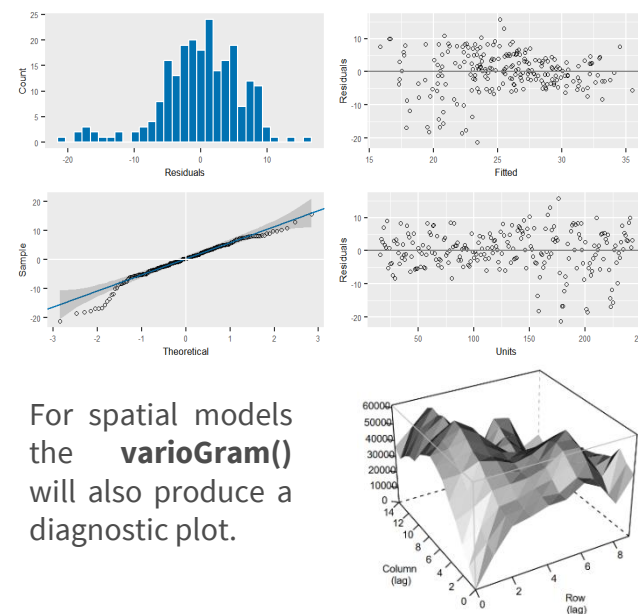
```
update(nin89.asr)
```

Extracting Output

Residual Plots

The **plot()** function used with an **asreml** object will produce the following four residual plots:

```
plot(nin89.asr)
```



For spatial models the **varioGram()** will also produce a diagnostic plot.

Residual Plots

The **summary()** function returns a list with a range of components, such as:

```
summary(nin89.asr)$loglik
```

```
[1] -454.4691
```

log-likelihood value

variance components table

```
summary(nin89.asr)$varcomp
```

	component	std.error	z.ratio	bound	%ch
	Rep!Rep	9.882911	8.792829	1.123974	P 0
	units!units	49.582368	5.458821	9.082981	P 0
	units!R	1.000000	NA	NA	F 0

```
summary(nin89.asr)$aic
```

```
[1] 912.9382
```

Akaike Information Criterion

```
summary(nin89.asr)$bic
```

```
[1] 919.1861
```

Bayesian Information Criterion

Coefficients/Solutions

The coefficients from the fixed and random parts of the model are available in the **coef.fixed** and **coef.random** or using **coef()**.

If Variety is a fixed effect = EBLUE

```
summary(nin89.asr, coef=TRUE)$coef.fixed
```

	solution	std.error	z.ratio
Variety_ARAPAHOE	29.4375	3.855683	7.634834
Variety_BRULE	26.0750	3.855683	6.762744
Variety_BUCKSKIN	25.5625	3.855683	6.629824
Variety_CENTURA	21.6500	3.855683	5.615088

If Variety is a random effect = EBLUP

```
summary(nin89.asr, coef=TRUE)$coef.random
```

	solution	std.error	z.ratio
Variety_ARAPAHOE	7.30E-05	0.01497209	0.004875042
Variety_BRULE	1.02E-05	0.01497209	0.000683157
Variety_BUCKSKIN	6.62E-07	0.01497209	4.42452E-05
Variety_CENTURA	-7.24E-05	0.01497209	-0.004833302

Wald Test

The basic tool for fixed effects inference is the **wald()**, and ANOVA-like table. The asymptotic version is obtained as:

```
wald(nin89.asr)
```

	Df	Sum of Sq	Wald statistic	Pr(Chisq)
(Intercept)	1	242.054	242.054	<2e-16 ***
Variety	55	48.152	48.152	0.7317
residual (MS)	1			

The version that obtains degrees of freedom is:

```
wald(nin89.asr, denDF = "numeric")
```

	df	denDF	F.inc	Pr
(Intercept)	1	3	242.1	0.00058
Variety	55	165	0.875	0.71185

Predict

The **predict()** method obtains an estimated or predicted values (means) for an element of interest.

```
rcb.pv <- predict(  
  object = nin89.asr, classify = "Variety")
```

```
rcb.pv$pvals
```

Variety	predicted.value	std.error	status
ARAPAHOE	29.4375	3.855687	Estimable
BRULE	26.0750	3.855687	Estimable
BUCKSKIN	25.5625	3.855687	Estimable
CENTURA	21.6500	3.855687	Estimable
CENTURK78	30.3000	3.855687	Estimable

The average standard error of difference (SED) between the predicted means is:

```
rcb.pv$saved # 4.979075
```

Functions of Variances

The option **vpredict()** allows to define functions of variance components and their SEs. These are represented by the strings "V1", "V2", ...

```
vpredict(model, H2 ~ V1/(V1+V2))
```

broad sense heritability

LRT Test

Function used to compare nested models based on the REML likelihood ratio test implemented with the function **lrt()**

```
lrt(model1, model2, boundary = TRUE)
```

	df	LR-statistic	Pr(Chisq)
model1/model2	1	-7.2562e-06	0.5