Avoiding the misuse of BLUP in behavioral ecology: II. Multivariate modelling for individual plasticity (ASReml-R tutorial)

Thomas M. Houslay & Alastair J. Wilson

January 2017

Introduction

Overview

This tutorial accompanies our 2017 Behavioral Ecology paper, “Avoiding the misuse of BLUP in behavioral ecology”. Here, we focus on **individual variation in plasticity**, studying (behavioural) traits measured repeatedly across some environmental gradient. We will:

- Use random regression to test for individual variation in plasticity under a reaction norm perspective;
- Add an additional response variable to the random regression, and test for an association between individual variation in intercepts/slopes and, for example, a single measure of lifetime fitness;
- Show how ‘character state’ models, set up in terms of environment-specific subtraits, can equally be used to model individual variation in plasticity;
- Add the fitness measure to the character state model to demonstrate why thinking about how selection acts on a trait in different environments may sometimes be a more intuitive way of looking at things than trying to estimate selection on “plasticity” (although both perspectives are valid).

In this version, we illustrate these models using the `R` interface for `ASReml`, which is commercial software available from VSNi. We have provided a separate tutorial for the free `R` package `MCMCglmm`, but note that `MCMCglmm` uses Bayesian methods while `ASReml` uses maximum likelihood (and so `ASReml` is likely to be more familiar to users of the `R` package `lme4`).

Updates and further tutorials associated with this paper can be found at https://tomhouslay.com/tutorials/.

Aims

As with the previous tutorial (‘I. Multivariate modelling for individual variation’), we assume readers are familiar with the general principles of specifying univariate mixed effects models, and using diagnostic plots to check that the fitted model does not violate assumptions of the linear model.

There are various papers that cover the fundamentals of mixed models for individual variation in behavioural plasticity, and we recommend Dingemanse *et al* (2010) ‘Behavioural reaction norms: animal personality meets individual plasticity’ and Brommer (2013) ‘Variation in plasticity of personality traits implies that the ranking of personality measures changes between environmental contexts: calculating the cross-environmental correlation’. We also provide further citations later in the tutorial for more specific topics.

We also use various methods for manipulating and visualising data frames using the `tidyverse` package (including `tidyr`, `dplyr`, `ggplot2` etc) — more details on their use can be found at http://r4ds.had.co.nz/.

In our tutorial, we aim to teach the following:

- Specifying, fitting and interpreting a random regression model;
- Adding a second response variable to a random regression model;

- Restructuring data in a format suitable for character state models;
- Fitting and interpreting character state models;
- Hypothesis testing at each stage using likelihood ratio tests.

Packages required

There are several packages that you must have installed in `R` prior to starting this tutorial:

- `asreml` (note that this should be provided by the vendor, VSNi)
- `lme4`
- `nadiv`
- `tidyverse`
- `broom`
- `gridExtra`

Behavioural plasticity

In this section, we will look at how one can test for and estimate individual variation in behavioural plasticity — which is also known as individual by environment interaction (IxE). A common approach to studying IxE is to utilise reaction norms which are typically assumed to be linear (first order function of the environmental variable, E). Under this perspective IxE is present if individuals differ in the slope of their phenotypic relationship to E. This variation in slopes (i.e. plasticities) can be modelled using random regression mixed models.

Individual variation in plasticity

As in our previous tutorial, we use wild haggis (*Haggis scoticus*) as our study organism.

Here, we have measured aggression in another population of wild haggis (using individually tagged males). Male haggis are territorial, and previous studies have suggested that aggressive behaviour tends to increase with the size of a rival male. For this experiment, we used model 'intruders' to test for individual variation in aggressive behaviour. These intruders were made up of 3 size classes: average male haggis size (calculated as the population mean), 1 standard deviation below the population mean, and 1 standard deviation above. We measured each focal male against each model size, and repeated this in 2 blocks (with test order randomised). The body size of the focal individual was measured at the beginning of each block. We also have a fitness proxy (a single measure of mating success for the season) for each male.

We want to test whether individuals vary in the extent of their aggression towards intruders, whether there really is a dependence of aggression on intruder size (on average), and – if so – whether the plastic response to intruder size varies among individuals (IxE). Finally, we want to test whether behavioural variation is associated with fitness as we'd expect if aggression is under natural selection.

Load libraries and inspect data

```r
library(lme4)
library(asreml)
library(tidyverse)
library(broom)
library(nadiv)
library(gridExtra)
```

Figure 1: A male haggis in the wild (thanks to Emma Wood, http://www.ewood-art.co.uk/)

```
df_plast <- read_csv("aggression.csv")
```

This data frame has 6 variables:

- Individual **ID**
- Experimental **Block**
- Individual **body_size**, as measured for each block
- The repeat number for each behavioural test, **assay_rep**
- Opponent size (**opp_size**), in standard deviations from the mean (i.e., -1,0,1)
- **aggression**, measured 6 times in total per individual (2 blocks of 3 tests)
- **fitness**, our measure of mating success, with a single value for each individual

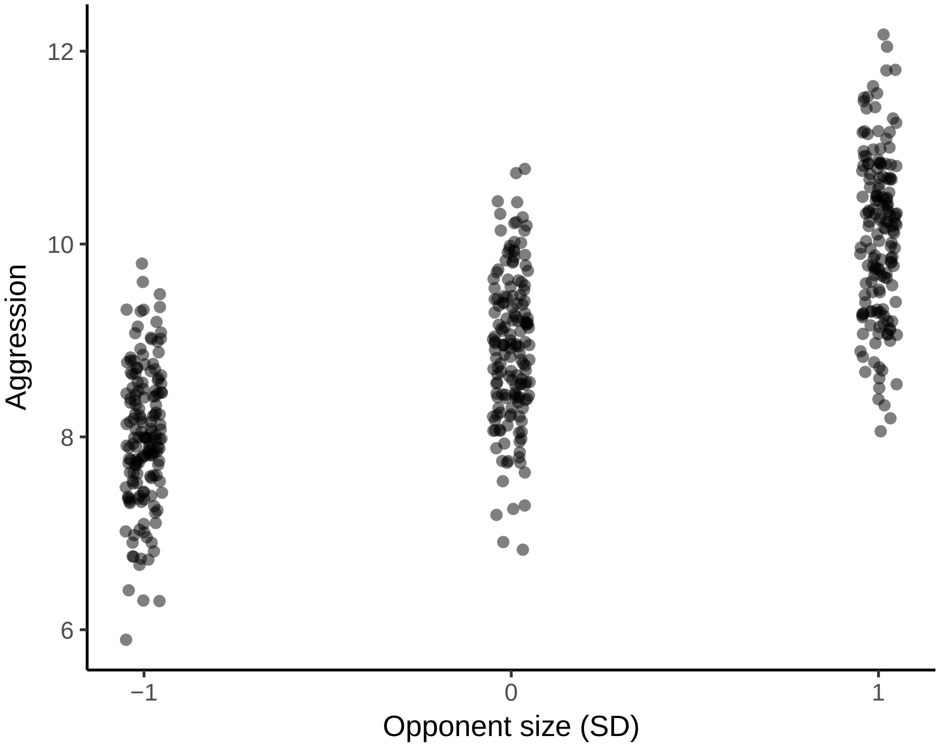
As in the previous tutorial, we want to convert fitness to **relative fitness** by dividing each individual's fitness value by the mean population fitness. We will not be using the raw fitness value at all, so we can simply overwite it in our data frame with this standardised version (of course, we do not change the csv file in which our collected data is stored, so we can always get the raw values again if needed):

```
df_plast <- df_plast %>%
  mutate(fitness = fitness/mean(fitness, na.rm=TRUE))
```

An initial plot of the phenotypic data indicates that - as expected - there is a general increase in aggression with opponent size (points are lightly jittered on the x-axis to show the spread of data a little better):

```
ggplot(df_plast, aes(x = opp_size, y = aggression)) +
  geom_jitter(alpha = 0.5,
              width = 0.05) +
```

```
  scale_x_continuous(breaks = c(-1,0,1)) +
  labs(x = "Opponent size (SD)",
       y = "Aggression") +
  theme_classic()
```



From the above plot, and also from a quick look at the population mean for aggression at each opponent size (below), we expect that our models should show a roughly linear increase in aggression (1 unit increase per standard deviation of opponent size).

```
df_plast %>%
  group_by(opp_size) %>%
  summarise(mean_aggr = mean(aggression)) %>%
  kable(digits = 2)
```

opp_size	mean_aggr
-1	8.00
0	8.91
1	10.09

Random intercepts model (lme4)

We can quickly run a mixed model with only random intercepts, to see how it fits the data. Here we fit fixed effects of opponent size (our predictor of interest), focal male body size (mean-centred and scaled), assay repeat (mean-centred), and experimental block. As we only had 2 blocks, we fit this as a fixed rather than random effect. Our random effect here is individual ID. So here we allow random intercepts to vary among males, and we also allow population-level plasticity (change in mean aggression with opponent size), but we do not model any IxE:

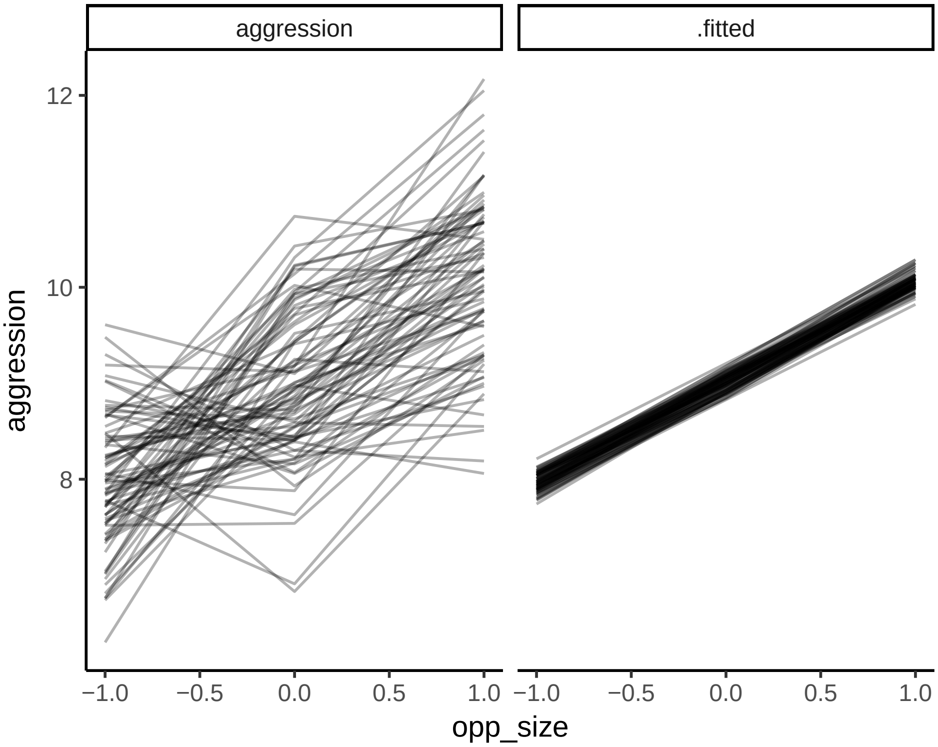
```
lmer_a <- lmer(aggression ~ opp_size + scale(body_size) +
                 scale(assay_rep, scale = FALSE) + block +
                  (1|ID),
                data = df_plast)
```

Diagnostic plots look fine:

```
plot(lmer_a)
qqnorm(residuals(lmer_a))
```

We then use the handy `augment` function from the `broom` package to get predictions for each of our observations. Below, we plot the raw data for each individual in one panel, with the fitted slopes in a second panel. Because we have 2 blocks as fixed effects, for ease of presentation we have selected only one of the blocks for this plot (if you like, you can check the other blocks to reassure yourself that there is little overall difference - another way would be to calculate predictions while averaging over block effects).

```
augment(lmer_a) %>%
  select(ID, block, opp_size, .fitted, aggression) %>%
  filter(block == -0.5) %>%
  gather(type, aggression,
         `.fitted`:aggression) %>%
  ggplot(., aes(x = opp_size, y = aggression, group = ID)) +
  geom_line(alpha = 0.3) +
  theme_classic() +
  facet_grid(.~type)
```



This illustrates the importance of using model predictions to see whether the model actually fits the individual-level data well or not — while the diagnostic plots looked fine, and the model captures mean plasticity, here we can see that the model really doesn't fit the actual data very well at all. The code below provides a different (and slightly more in-depth) look at this same combination of fitted slope / real data, and indicates that the fitted slopes systematically under- (eg, `ID_42`, `ID_43`) and over-estimate (eg, `ID_5`, `ID_25`) plasticity in aggression at the individual level (we have not shown the figure as it consists of 80 panels!).

```
# Create 'dummy' data frame for prediction
# - all IDs, all opponent sizes,
# - block set at 0 (as blocks are coded as -0.5, 0.5)
# - mean body size and assay repeats
df_ri_ind <- expand(df_plast,
                    ID, opp_size,
                    block = 0,
                    body_size = mean(body_size),
                    assay_rep = mean(assay_rep))

# Get predicted values based on RR model and dummy data frame
#  and using random effects structure as in the model
df_ri_ind$fit <- predict(lmer_a, newdata = df_ri_ind, re.form = NULL)

# Plot predictions and overlay original data points
ggplot(df_ri_ind, aes(x = opp_size, y = fit, group = ID)) +
  geom_line() +
  geom_point(data = df_plast,
             aes(y = aggression),
             alpha = 0.3) +
  scale_x_continuous(breaks = c(-1,0,1)) +
  theme_classic() +
  facet_wrap(~ID)
```

Now, let's move on and test whether random slopes provide a better fit to the data…

Random regression (lme4)

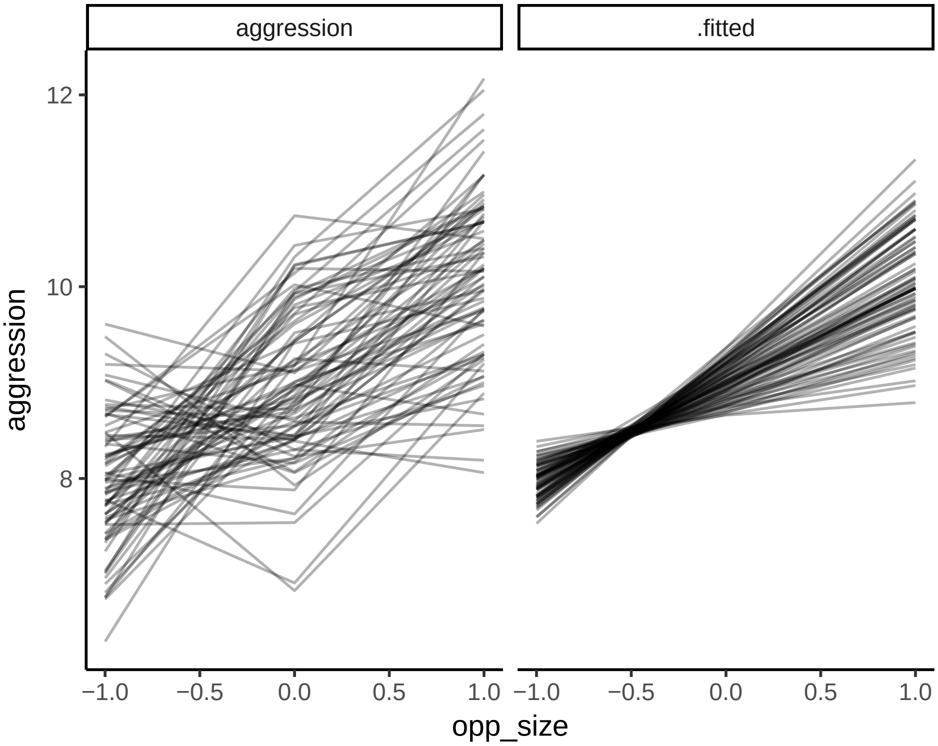
```
lmer_a_rr <- lmer(aggression ~ opp_size + scale(body_size) +
                  scale(assay_rep, scale = FALSE) + block +
                   (opp_size|ID),
                 data = df_plast)
```

```
plot(lmer_a_rr)
qqnorm(residuals(lmer_a_rr))
```

The random regression model also fits the data well — let's go ahead and check our predicted fit (again, looking at a single block - simply change the `filter` specification to check predictions with the other):

```
augment(lmer_a_rr) %>%
  select(ID, block, opp_size, .fitted, aggression) %>%
  filter(block == -0.5) %>%
  gather(type,aggression,
         `.fitted`:aggression) %>%
```

```
  ggplot(., aes(x = opp_size, y = aggression, group = ID)) +
  geom_line(alpha = 0.3) +
  theme_classic() +
  facet_grid(.~type)
```



This certainly seems better, as it is capturing both mean plasticity and variation at the individual level. Again, we have also provided code for a more in-depth look at individual predictions against the raw data:

```
# Create 'dummy' data frame for prediction
# - all IDs, all opponent sizes,
# - block set at 0 (as blocks are coded as -0.5, 0.5)
# - mean body size and assay repeats
df_rr_ind <- expand(df_plast,
                    ID, opp_size,
                    block = 0,
                    body_size = mean(body_size),
                    assay_rep = mean(assay_rep))

# Get predicted values based on RR model and dummy data frame
#  and using random effects structure as in the model
df_rr_ind$fit <- predict(lmer_a_rr, newdata = df_rr_ind, re.form = NULL)

# Plot predictions and overlay original data points
ggplot(df_rr_ind, aes(x = opp_size, y = fit, group = ID)) +
  geom_line() +
  geom_point(data = df_plast,
             aes(y = aggression),
             alpha = 0.3) +
```

```
  scale_x_continuous(breaks = c(-1,0,1)) +
  theme_classic() +
  facet_wrap(~ID)
```

Compared to the individual-level plots of the random intercepts model, this appears to be doing a much better job of fitting the data.

We can test the improvement of the model fit using the overloaded `anova` function in R to perform a likelihood ratio test (LRT):

```
anova(lmer_a_rr, lmer_a)
```

```
## refitting model(s) with ML (instead of REML)
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
lmer_a	7	1128.709	1157.925	-557.3544	1114.709	NA	NA	NA
lmer_a_rr	9	1070.269	1107.833	-526.1345	1052.269	62.43989	2	0

We can see here that the LRT uses a chi-square test with 2 degrees of freedom, and indicates that the random slopes model shows a statistically significant improvement in model fit. The 2df are because there are two additional (co)variance terms estimated in the random regression model: a variance term for individual slopes, and the covariance (or correlation) between the slopes and intercepts. Let's look at those values, and also the fixed effects parameters, via the model summary:

```
summary(lmer_a_rr)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## aggression ~ opp_size + scale(body_size) + scale(assay_rep, scale = FALSE) +
##     block + (opp_size | ID)
##    Data: df_plast
##
## REML criterion at convergence: 1074.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.04930 -0.59780 -0.02004  0.59574  2.68010
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ID       (Intercept) 0.05043  0.2246
##           opp_size    0.19166  0.4378   0.96
##  Residual             0.42817  0.6543
## Number of obs: 480, groups:  ID, 80
##
## Fixed effects:
##                                 Estimate Std. Error t value
## (Intercept)                      9.00181    0.03902  230.71
## opp_size                         1.05033    0.06123   17.15
## scale(body_size)                 0.02725    0.03377    0.81
```

```
## scale(assay_rep, scale = FALSE) -0.04702    0.03945   -1.19
## block                          -0.02169    0.05973   -0.36
##
## Correlation of Fixed Effects:
##              (Intr) opp_sz scl(_) s(_s=F
## opp_size      0.495
## scl(bdy_sz)   0.000  0.000
## s(_,s=FALSE   0.000 -0.064 -0.006
## block         0.000  0.000  0.002  0.000
```

From the fixed effects, we see that there is a strong effect of opponent size on aggression (1.05 SE 0.06, t = 17.15) — so there is population-level plasticity at a rate anticipated from our earlier plots of raw data, whereby aggression increases (on average) at a rate of 1 unit per standard deviation increase in opponent size. Significance of fixed effects can be determined using likelihood ratio tests (you should find that testing `opp_size` returns $P<0.001$). The other fixed effects in this model — individual body size, the order of assays, and block effects — are small and non-significant.

Interpreting the random effects requires some care. Remember that the intercept variance is the among-individual variance in aggression where x (here, opponent size) = 0. Here this corresponds to an average sized opponent because of the way we scaled the environmental covariate. This is very important! **A different scaling would change the number (but not the model predictions!)**. There is also a highly positive correlation between intercepts and slopes **at the intercept x = 0, i.e., where opponent size is at the mean population value**. This positive correlation indicates that individuals with more positive intercept deviations also have more positive slope deviations **at x = 0**.

When looking at our (above) plot of predicted individual slopes, it is clear that the intercept-slope correlation is highly dependent on the positioning of the intercept. If the intercept was positioned elsewhere then we might get a very different intercept-slope correlation. This is absolutely as it should be — for example, if we had called our opponent sizes 1:3 then x=0 would be off to the far left, and the intercept-slope correlation would be closer to -1 as it would be the least positive intercept deviations that had the most positive slope deviations at that value of x. Failure to understand the dependence of the random effect parameters on the scaling and centring of x means a high risk of interpreting the biological meaning of your model incorrectly.

Similarly, if there is IxE we have to be careful about drawing conclusions from the among-individual variation in intercepts in our model summary. To reiterate, it is the variance among-individuals at x=0, but in the presence of IxE (i.e., individual slopes are allowed to vary) then the among-individual variation changes as a function of opponent size. **So, it is not correct to view variation in intercepts as behavioural variation that is "independent of the environment" as is sometimes stated**. Furthermore, consider again the hypothetical situation above where we coded our opponent sizes as 1:3 (rather than -1:1). In such a situation, the variation in intercepts would still be calculated at x = 0, so not even in the range of where we collected the data!

From the above plots, it is clear that (in the case of our data) here the among-individual variation in aggression is greater at large opponent size than at small opponent size. Later in the tutorial we will look at different methods that enable us to investigate changes in among-individual variance more explicitly. A more thorough discussion of such issues when modelling IxE can be found in Nussey *et al* (2007) 'The evolutionary ecology of individual phenotypic plasticity in wild populations'.

This section was simply a quick reminder of how a random regression works – and some important notes on interpretation – using what's probably a familiar setup. For more advanced modelling techniques we will use `ASReml`, so let's start by seeing how to run this same random regression model using that.

Random regression (ASReml)

Fitting a random regression (also known as 'random slopes' models) in `ASReml` is very similar to in `lme4`, but the `random` term is slightly more involved because there are more options for fitting different covariance structures among the random effects. With this increased flexibility comes a need to be more exact in our syntax about what structure we want to use!

The random regression structure begins with `str`, within which we specify that we want to model both random intercepts for individuals (`ID`) and random slopes of opponent size for each individual (`ID:opp_size`). The `us` keyword indicates that we want to fit an unstructured covariance matrix for these effects (i.e., we fit variance in intercepts, variance in slopes, and the covariance between them). Within the `us` specification we can also provide initial values for our (co)variances terms: here we give starting values of unit variance for both intercept and slope variance, and a small positive value for their covariance. Finally, `id` requires that we specify the number of levels within our random effect (here 80, as we have 80 individuals). We then simply model residuals in `rcov` as `units`.

```
asr_A_RR <- asreml(aggression ~ opp_size +
                     scale(assay_rep, scale=FALSE) +
                     scale(body_size) +
                     block,
                  random =~ str(~ID + ID:opp_size,
                               ~us(2, init = c(1,0.1,1)):id(80)),
                  rcov =~ units,
                  data = df_plast,
                  maxiter = 200)
```

Take a look at diagnostic plots for the model fit:

```
plot(asr_A_RR)
```

As before, we can also take a look at the fixed effects results if desired:

```
summary(asr_A_RR, all = T)$coef.fixed
```

We could also look at the Wald F-tests, if we wanted statistics and p-values to indicate that there is a significant effect of opponent size on aggression:

```
wald.asreml(asr_A_RR, ssType="conditional", denDF="numeric")
```

Again, fixed effects show that there is a significant effect of opponent size on aggression at the population level, whereby for every standard deviation increase in opponent size we see just over 1 unit of increase in aggression.

Of course, because we want to know about individual variation in behaviour, we are more interested in the random effects of our model. Because we used the `us` covariance structure, we get the variance in intercepts (line 1, "ID+ID:opp_size!us(2).1:1"), variance in slopes (line 3, "2:2"), and covariance between them (line 2, "2:1"). Note that the numbers here relate to position in the 2x2 covariance matrix we specified with our random effects structure: the first element is among-individual variance in intercepts and the second is among-individual variance in slopes (such that the off-diagonals would be the covariance between them). We also get an estimate of the residual (within-individual) variance (line 4, "R!variance"):

```
summary(asr_A_RR)$varcomp
```

```
##                                 gamma  component  std.error   z.ratio
## ID+ID:opp_size!us(2).1:1 0.1177788 0.05042913 0.02027570  2.487171
## ID+ID:opp_size!us(2).2:1 0.2209016 0.09458301 0.02400788  3.939666
## ID+ID:opp_size!us(2).2:2 0.4476246 0.19165854 0.04832198  3.966281
```

```
## R!variance              1.0000000 0.42816798 0.03395301 12.610605
##                         constraint
## ID+ID:opp_size!us(2).1:1   Positive
## ID+ID:opp_size!us(2).2:1   Positive
## ID+ID:opp_size!us(2).2:2   Positive
## R!variance                 Positive
```

As above, we can also test whether the random slopes model provides a better fit to the data by fitting a random intercepts model, then using a likelihood ratio test (with 2 degrees of freedom, as the random slopes model fits not only random slopes but also the random intercepts-slopes covariance):

```
asr_A_RI <- asreml(aggression ~ opp_size +
                     scale(assay_rep, scale=FALSE) +
                     scale(body_size) +
                     block,
                   random =~ ID,
                   rcov =~ units,
                   data = df_plast,
                   maxiter = 200)
```

```
pchisq(2*(asr_A_RR$loglik - asr_A_RI$loglik),
       2, lower.tail = FALSE)
```

```
## [1] 3.241026e-14
```

Note that the chi-square statistic is simply twice the difference in model log-likelihoods:

```
2*(asr_A_RR$loglik - asr_A_RI$loglik)
```

```
## [1] 62.1206
```

So, just as we found fitting the model with lmer, this result from ASreml fits tells us there is significant among-individual variation in behavioural plasticity (IxE) — i.e., individuals change their aggression at different rates in response to opponent size.

It is sometimes easier to understand covariances by rescaling them as correlations: we can use the `pin` function to convert the among-individual covariance between intercepts and slopes to a correlation (with associated standard error) using the standard formula (i.e., dividing the covariance by the product of the squared variances):

```
nadiv:::pin(asr_A_RR, cor_is ~ V2/(sqrt(V1)*sqrt(V3)))
```

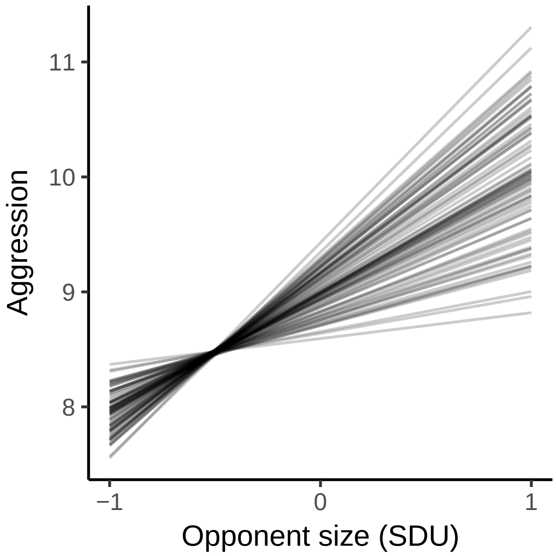
```
##         Estimate        SE
## cor_is 0.9620736 0.1773967
```

This indicates that the among-individual correlation between individual intercepts and slopes is positive — the correlation estimate is the same as our lme4 model — although also note that the standard error is fairly large.

Just as with the lmer model, we can use a `predict` function to extract individual-level predictions and plot them. `ASReml-R`'s predict function can be a little tricky to use — here we have given it:

- The name of the model object
- The variables we want used in making predictions (other variables will be averaged over)
- The values to be used for predicting opponent size

```
# Retrieve individual predictions
#  from ASReml random regression model
#  and reform into data frame for plotting
df_asr_RR_pred <- as.data.frame(predict(asr_A_RR,
                                        classify = "opp_size:ID",
                                        levels = list("opp_size" =
                                                        c(opp_size = -1:1)))$prediction$pvals)

ggplot(df_asr_RR_pred, aes(x = opp_size,
                           y = predicted.value,
                           group = ID)) +
  geom_line(alpha = 0.2) +
  scale_x_continuous(breaks = c(-1,0,1)) +
  labs(x = "Opponent size (SDU)",
       y = "Aggression") +
  theme_classic()
```



Bivariate model with random regression on one trait

Having established that there is among-individual variation in behavioural plasticity, we want to now test whether there is an association between variation in aggression (intercept and/or slope) and 'fitness' (measured as mating success — recall also that we standardised this to relative fitness earlier in the tutorial).

There are several different ways to code this bivariate model, and we'll just look at one option. In general, if we had two response variables that both had repeat observations on individuals and might both be functions of the x variable (opponent size), we might want `ASReml` to try to fit a random slope and intercept to each response variables. The setup of a covariance matrix for a bivariate random regression like this in `ASReml` is as follows (where T1 and T2 represent our two response trait terms):

	$T1_{int}$	$T2_{int}$	$T1_{slope}$	$T2_{slope}$
$T1_{int}$	V	COV	COV	COV

	T1_{int}	T2_{int}	T1_{slope}	T2_{slope}
T2_{int}	COV	V	COV	COV
T1_{slope}	COV	COV	V	COV
T2_{slope}	COV	COV	COV	V

However, here we do not measure the second response variable (fitness) at different values of x, so that would not work. While a 'clean' way would be to simply specify the random slope for aggression only, the model does not seem to run with this specification. We therefore have to constrain the variance in random slopes for fitness to a very low positive number (such that it is effectively zero), and all of its correlations to zero, such that:

	T1_{int}	T2_{int}	T1_{slope}	T2_{slope}
T1_{int}	V	COV	COV	0
T2_{int}	COV	V	COV	0
T1_{slope}	COV	COV	V	0
T2_{slope}	0	0	0	~0

In addition, fitness is measured only once, so we need to constrain the within-individual variance such that it is effectively zero.

Our first task is therefore to set up these constraints within the starting values that we include in the random effects structure of the model specification. See the previous tutorial for more detailed description of setting up model constraints, but briefly we use `U` to specify 'unconstrained' variances/correlations, and `F` to specify those that we have fixed at their starting value.

Random effects constraints: we are going to use a correlation structure here, `corgh`, which fits a very similar model to `us` but with correlations (rather than covariances) on the off-diagonals. When we use a correlation structure (such as `corgh`), we specify the correlations first, then the variance terms. So, our starting values consist of 6 correlations, followed by 4 variances. Following the structure in the example matrix above, we set 3 low unconstrained correlations, 3 correlations fixed at 0 (all those involving the slope of the second response variable), 3 unconstrained variance terms (trait 1 intercept, trait 2 intercept, trait 1 slope), and a final variance term fixed at a very small positive number (trait 2 slope – which doesn't exist, but variances must be positive and non-zero).

```
init_cor_RR_fit <- c(0.1,
                     0.1,0.1,
                     0,0,0,
                     1,1,1,1e-08)
names(init_cor_RR_fit) <- c("U",
                            "U","U",
                            "F","F","F",
                            "U","U","U","F")
```

Residual variation constraints: we use the `idh` structure as we do not fit a within-individual covariance. The first element of the residuals is our repeated measure, so this is set to be unconstrained. The second element is the observation made only once per individual — meaning that there is effectively no residual variation. We therefore set the residual for this trait to a very small positive number (again, because variances must be positive and non-zero).

```
init_res_RR_fit <- c(0.1, 1e-08)
names(init_res_RR_fit) <- c("U","F")
```

With the model itself, we standardise aggression by scaling it (to make the multivariate model easier to fit), and we have already standardised fitness as relative fitness earlier on.

The `at` keyword specifies fixed effects as relating to only one of our response variables — here, we have fixed effects of opponent size, block, repeat and body size for aggression (but not for fitness).

We set up our `random` effects in a similar way to the univariate random regression, but using `corgh` instead of `us`, and telling the model that we have 4 variance terms (i.e., 2 random intercepts variances, and 2 random slopes variances).

Finally, we fit residual variances for each of our response variables (remember that we have set residual variance in fitness to be essentially zero).

```
asr_A_RR_fit <- asreml(cbind(scale(aggression),
                             fitness) ~ trait +
                         at(trait,1):opp_size +
                         at(trait,1):block +
                         at(trait,1):scale(assay_rep, scale=FALSE) +
                         at(trait,1):scale(body_size),
                   random =~ str(~trait:ID + trait:ID:opp_size,
                                 ~corgh(4, init = init_cor_RR_fit):id(80)),
                   rcov =~ units:idh(trait, init = init_res_RR_fit),
                   data = df_plast,
                   maxiter = 500)

plot(residuals(asr_A_RR_fit) ~ fitted(asr_A_RR_fit))
qqnorm(residuals(asr_A_RR_fit))
hist(residuals(asr_A_RR_fit))
```

We can take a look at the variance components and their standard errors for this model:

```
summary(asr_A_RR_fit)$varcomp[,c("component","std.error")]
```

```
##                                                         component
## trait:ID+trait:ID:opp_size!corgh(4).2:!corgh(4).1.cor 0.51710326
## trait:ID+trait:ID:opp_size!corgh(4).3:!corgh(4).1.cor 0.96194717
## trait:ID+trait:ID:opp_size!corgh(4).3:!corgh(4).2.cor 0.46237199
## trait:ID+trait:ID:opp_size!corgh(4).4:!corgh(4).1.cor 0.00000000
## trait:ID+trait:ID:opp_size!corgh(4).4:!corgh(4).2.cor 0.00000000
## trait:ID+trait:ID:opp_size!corgh(4).4:!corgh(4).3.cor 0.00000000
## trait:ID+trait:ID:opp_size!corgh(4).1                 0.03798792
## trait:ID+trait:ID:opp_size!corgh(4).2                 0.02614454
## trait:ID+trait:ID:opp_size!corgh(4).3                 0.14365560
## trait:ID+trait:ID:opp_size!corgh(4).4                 0.00000001
## R!variance                                            1.00000000
## R!trait.aggression                                    0.32089123
## R!trait.fitness                                       0.00000001
##                                                         std.error
## trait:ID+trait:ID:opp_size!corgh(4).2:!corgh(4).1.cor 0.156675825
## trait:ID+trait:ID:opp_size!corgh(4).3:!corgh(4).1.cor 0.176768135
## trait:ID+trait:ID:opp_size!corgh(4).3:!corgh(4).2.cor 0.117984902
## trait:ID+trait:ID:opp_size!corgh(4).4:!corgh(4).1.cor          NA
## trait:ID+trait:ID:opp_size!corgh(4).4:!corgh(4).2.cor          NA
## trait:ID+trait:ID:opp_size!corgh(4).4:!corgh(4).3.cor          NA
## trait:ID+trait:ID:opp_size!corgh(4).1                 0.015235076
```

```
## trait:ID+trait:ID:opp_size!corgh(4).2                  0.004159899
## trait:ID+trait:ID:opp_size!corgh(4).3                  0.036217335
## trait:ID+trait:ID:opp_size!corgh(4).4                           NA
## R!variance                                                      NA
## R!trait.aggression                                     0.025445583
## R!trait.fitness                                                 NA
```

The summary of variance components gives us estimates (and standard errors) of the 6 among-individual correlations, followed by the 4 variance components. The `R!variance` term can be ignored, and instead focus on the residual variance terms given for each trait (aggression and fitness).

A good thing to check first is that the among-individual correlation between intercepts and slopes for aggression is (approximately) the same as we estimated in our earlier univariate random regression model. As we can see from our example matrix above, this is the correlation between variance components 1 and 3 (here given in line 2 of the summary, and denoted `corgh(4).3:!corgh(4).1.cor`). This does look the same, which is as it should be!

The correlation between elements 1 and 2 (line 1 of the summary, `corgh(4).2:!corgh(4).1.cor`) is the among-individual correlation between fitness and variation in **intercepts** for aggression — here we find a **positive** correlation. We also find a **positive** among-individual correlation between fitness and variation in **slopes** for aggression (line 3, `corgh(4).3:!corgh(4).2.cor`).

Again, recall that variation in intercepts is among-individual variance for aggression at x=0 in a random regression model, so the correlation between fitness and intercept is interpetable as the fitness-aggressiveness correlation in this ‘environment’.

We now test for an overall association between individual variation in aggression and in fitness, by putting some additional constraints on our model. Here we specify that both the correlations between aggression intercepts:fitness and aggression slopes:fitness are set to 0:

```
init_cor_RR_fit_test <- c(0,0.1,0,
                          0,0,0,
                          1,1,1,
                          0)
names(init_cor_RR_fit_test) <- c("F","U","F",
                                 "F","F","F",
                                 "P","P","P",
                                 "F")

asr_A_RR_fit_test <- asreml(cbind(scale(aggression),
                                  fitness) ~ trait +
                              at(trait,1):opp_size +
                              at(trait,1):block +
                              at(trait,1):scale(assay_rep, scale=FALSE) +
                              at(trait,1):scale(body_size),
                            random =~ str(~trait:ID + trait:ID:opp_size,
                                          ~corgh(4, init = init_cor_RR_fit_test):id(80)),
                            rcov =~ units:idh(trait, init = init_res_RR_fit),
                            data = df_plast,
                            maxiter = 500)
```

We then use a likelihood ratio test with 2 degrees of freedom to determine whether modelling an association between aggression and fitness provides a significantly better fit to the data:

```
pchisq(2*(asr_A_RR_fit$loglik - asr_A_RR_fit_test$loglik),
       2, lower.tail = FALSE)
```

```
## [1] 0.0007922543
```

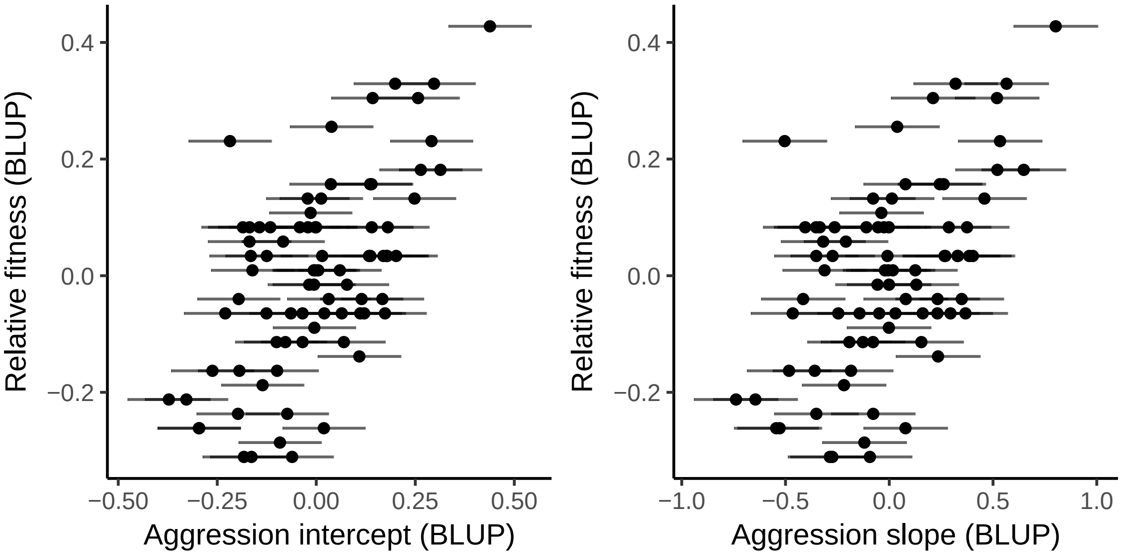
So we do find a significant association between variation in aggression and in fitness. Below is the variance-covariance matrix (with variances on the diagonal, correlations above, and covariances below):

	Aggression	Fitness	Aggression:Opp	Fitness:Opp
Aggression	0.04	0.52	0.96	0
Fitness	0.02	1	0.46	0
Aggression:Opp	0.07	0.03	0.14	0
Fitness:Opp	0	0	0	~0

We can also visualise this result by extracting the BLUPs for each individual and plotting them. Note that as in the previous tutorial, we think this is a reasonable use of BLUP. We are not running statistical analyses on them, we are using the BLUPs (ie the model predictions) only as a visual aid for the interpretation of a statistical model we have fitted. We'll also make sure we show the uncertainty on those predictions here:

```
# Extract BLUPs and standard errors for all traits
df_agfit_ranef <- data_frame(Trait = attr(summary(asr_A_RR_fit,all=TRUE)$coef.random, "dimnames")[[1]],
                             Value = summary(asr_A_RR_fit,all=TRUE)$coef.random[,"solution"],
                             SE = summary(asr_A_RR_fit,all=TRUE)$coef.random[,"std error"])%>%
  separate(Trait, c("Trait","ID","Type"), sep = ":", fill = "right") %>%
  mutate(Type = ifelse(is.na(Type), "Intercept", "Slope")) %>%
  unite(Trait,
        Trait, Type)

# Make separate data frames for SEs for
# (a) aggression intercepts and (b) aggression slopes,
# as otherwise it gets a bit problematic with the data wrangling

df_aggr_int_se <- df_agfit_ranef %>%
  filter(Trait == "trait_aggression_Intercept") %>%
  select(ID, SE)

df_aggr_slope_se <- df_agfit_ranef %>%
  filter(Trait == "trait_aggression_Slope") %>%
  select(ID, SE)


# Create data frames of
#  (a) intercepts and fitness,
#  (b) slopes and fitness.

df_agfit_int <- df_agfit_ranef %>%
  select(-SE) %>%
  filter(Trait %in% c("trait_aggression_Intercept",
                      "trait_fitness_Intercept")) %>%
  spread(Trait, Value)
```

```r
df_agfit_slope <- df_agfit_ranef %>%
  select(-SE) %>%
  filter(Trait %in% c("trait_aggression_Slope",
                      "trait_fitness_Intercept")) %>%
  spread(Trait, Value)


# Add standard errors from appropriate data frames

df_agfit_int <- left_join(df_agfit_int,
                          df_aggr_int_se,
                          by = "ID")

df_agfit_slope <- left_join(df_agfit_slope,
                          df_aggr_slope_se,
                          by = "ID")

# Create both figures

ggplot(df_agfit_int,
       aes(x = trait_aggression_Intercept,
           y = trait_fitness_Intercept)) +
  geom_point() +
  geom_errorbarh(aes(xmin = trait_aggression_Intercept - SE,
                     xmax = trait_aggression_Intercept + SE),
                 height = 0,
                 alpha = 0.6) +
  labs(x = "Aggression intercept (BLUP)",
       y = "Relative fitness (BLUP)") +
  theme_classic()


ggplot(df_agfit_slope,
       aes(x = trait_aggression_Slope,
           y = trait_fitness_Intercept)) +
  geom_point() +
  geom_errorbarh(aes(xmin = trait_aggression_Slope - SE,
                     xmax = trait_aggression_Slope + SE),
                 height = 0,
                 alpha = 0.6) +
  labs(x = "Aggression slope (BLUP)",
       y = "Relative fitness (BLUP)") +
  theme_classic()
```



We can see that those with higher average aggression (at the intercept) tend to have higher fitness, but also those that have higher slope values have higher fitness. These are often characterised as (for example) correlations between fitness and average aggression, and fitness and plasticity. However, interpreting (and generalising) this kind of result is actually more difficult than it seems. For instance, could we infer that plasticity is under selection? **Maybe, but not necessarily**. For example, we know that slopes and intercepts are strongly positively correlated (at x=0), so we would at least need to try and separate out the "direct" effects of each "trait" (i.e. intercept and slope) on fitness. This could be done, following the conceptual approach of classical selection analysis (e.g. Lande and Arnold 1983) by calculating the partial regressions (interpretable as selection gradients) on intercept and slope on fitness.

We also know that significant among-individual variation in slopes really means that the among-individual variation in aggression changes as a function of our x variable (here, opponent size): in other words, individual-by-environment interactions are occurring. Furthermore, we know that the intercept-fitness covariance could be used as a measure of selection on aggressiveness at x=0. So instead of trying to separate selection on the trait (aggressiveness) mean from selection on plasticity, an alternative way of thinking about this scenario is to ask **how fitness relates to aggression at different values of opponent size**.

In other words, how does selection on a trait change with environment? Getting a more concrete interpretation of such results is where random regression models start to fall down a little. However, if our environmental variable (here, opponent size) can be considered a set of distinct environments, we can move into an alternative (and in our view better for many scenarios) way of modelling IxE…

Character state models

In fact, reaction norms as modelled here by 'random regression' can be viewed as (normally) reduced parameter versions of what's called a **'character state'** model — for more details on this, we recommend the chapter 'Quantifying Genotype-by-Environment Interactions in Laboratory Systems' by Roff & Wilson in Hunt & Hosken's 2014 textbook 'Genotype-by-Environment Interactions and Sexual Selection'. A character state approach posits that observations for each 'environment' (in this case, opponent size) are best modelled as distinct environment-specific sub-traits using multivariate analyses. With a lot of environments you have a lot of subtraits, so a lot of model parameters, and thus need a lot of data! This is one of the reasons why random regression is popular - you reduce a potentially infinite number of sub traits to a line described by just two parameters (intercept and slope). However, this saving comes at the cost of assuming a reaction norm shape (and if your assumptions are poor you cannot expect your model to fit well) and sometimes – as we have argued above – a lack of easy interpretation of your results.

Luckily, our experimental design here enables us to readily use a character state model. We have only 3 environments and plenty of data — moreover, because we have repeated measures within each 'environment', we can partition among- from within-individual (co)variation across each (see Brommer (2013) 'On between-individual and residual (co)variances in the study of animal personality' for further discussion of this topic).

We'll start by fitting just our aggression data in a character state model, meaning that we will estimate the among-individual variance in (and covariance between) each of our 'sub-traits': aggression against opponents 1SD below the mean body size, aggression against opponents of mean body size, and aggression against opponents 1SD above mean body size.

Modelling IxE for aggression

We need to do some rearranging of our data frame to structure it for this type of model. We want to end up with an individual's aggression values for each 'environment' (for a single block) on a single row in the data frame, with each 'sub-trait' in a different column (let's call these sub-traits `agg_S`, `agg_M`, and `agg_L`). Each row will also have the observation of block, and the opponent body size measurement for that block.

In this case, we have already done the 'data wrangling' for you, and provided the rearranged data frame in a new CSV file that we can load up straight away. However, at the end of this tutorial you will find the code for how we did this from the original data.

```
df_plast_CS <- read_csv("aggression_CS.csv")
```

Check this new data frame to understand the format and how it relates to our original data frame:

```
head(df_plast_CS)
```

```
## # A tibble: 6 × 8
##      ID block body_size agg_L agg_M agg_S opp_order  fitness
##   <chr> <dbl>     <dbl> <dbl> <dbl> <dbl>     <chr>    <dbl>
## 1  ID_1  -0.5     20.58 10.67 10.22  7.02     S_L_M 1.033846
## 2  ID_1   0.5     20.73 10.51  8.95  8.44     M_L_S       NA
## 3 ID_10  -0.5     28.32 10.81  9.43  7.73     L_S_M 0.960000
## 4 ID_10   0.5     28.80 10.67  9.46  8.08     L_S_M       NA
## 5 ID_11  -0.5     22.89  9.77  7.63  8.06     S_L_M 1.083077
## 6 ID_11   0.5     23.62 10.84  8.23  8.16     S_L_M       NA
```

We will first use a **trivariate model** to investigate (co)variances for our three 'sub-traits' (aggression for each opponent size class). We bind these as three response variables, interacting the `trait` keyword with the fixed effects of interest. Doing this means we estimate the effect of (for example) an individual's body size on aggression at each opponent size.

We use the `us` structure to estimate variances and covariances at both the among-individual (`random`) and within-individual/residual (`rcov`) level, then use diagnostic plots to check the model fit:

```
asr_A_CS <- asreml(cbind(agg_S,
                         agg_M,
                         agg_L) ~ trait +
                     trait:scale(body_size) +
                     trait:block +
                     trait:opp_order,
                   random =~ ID:us(trait, init = c(0.1,
```

```
                                                    0.1,0.1,
                                                    0.1,0.1,0.1)),
                   rcov =~ units:us(trait),
                   data = df_plast_CS,
                   maxiter = 200)


plot(residuals(asr_A_CS) ~ fitted(asr_A_CS))
qqnorm(residuals(asr_A_CS))
hist(residuals(asr_A_CS))
```

The summary of fixed effects terms will show you that the mean level of aggression at each opponent size (the intercept for each trait: `trait_agg_S`, `trait_agg_M`, `trait_agg_L`) seems to increase in a roughly linear fashion, which is what we expected from our initial plot of the data and from the random regression models.

```
summary(asr_A_CS, all=T)$coef.fixed
```

You'll see that there are various values for order now because it is a many-level factor — we can look at the Wald F-tests for statistics and p-values on the fixed effects to see whether they are having significant overall effects on our multi-response trait:

```
wald.asreml(asr_A_CS, ssType="conditional", denDF="numeric")
```

The variance components give among- and within-individual (co)variances for all traits:

```
summary(asr_A_CS)$varcomp[,c("component","std.error")]
```

```
##                                component  std.error
## ID:trait!trait.agg_S:agg_S   0.192960004 0.06321878
## ID:trait!trait.agg_M:agg_S -0.168519655 0.05085582
## ID:trait!trait.agg_M:agg_M   0.245594379 0.07096321
## ID:trait!trait.agg_L:agg_S -0.151990274 0.05660805
## ID:trait!trait.agg_L:agg_M   0.158418600 0.06375045
## ID:trait!trait.agg_L:agg_L   0.312548409 0.09125332
## R!variance                   1.000000000         NA
## R!trait.agg_S:agg_S          0.318089964 0.05198129
## R!trait.agg_M:agg_S          0.010362395 0.03695485
## R!trait.agg_M:agg_M          0.322379906 0.05248302
## R!trait.agg_L:agg_S         -0.009311624 0.04168401
## R!trait.agg_L:agg_M          0.159240477 0.04569265
## R!trait.agg_L:agg_L          0.405941964 0.06679539
```

If we want, we can easily test for among-individual covariance between our 3 sub-traits. As a "global" test we can fit a model with the `idh` structure in the random effects (which allows an among-individual variance for each trait but no covariance between traits in this part of the model), then use a likelihood ratio test to assess the two models:

```
asr_A_CS_idh <- asreml(cbind(agg_S,
                             agg_M,
                             agg_L) ~ trait +
                         trait:scale(body_size) +
```

```
                          trait:block +
                          trait:opp_order,
                        random =~ ID:idh(trait, init = c(0.1,0.1,0.1)),
                        rcov =~ units:us(trait),
                        data = df_plast_CS,
                        maxiter = 200)
```

```
pchisq(2*(asr_A_CS$loglik - asr_A_CS_idh$loglik),
       3, lower.tail = FALSE)
```

```
## [1] 0.0002039034
```

So we do find significant among-individual covariance structure among opponent size-specific aggression subtraits. This analysis doesn't tell us what drives this signal however, so it's useful to look at the among-individual correlations between each of these size classes:

```
nadiv:::pin(asr_A_CS, cor_S_M ~ V2/(sqrt(V1)*sqrt(V3)))
nadiv:::pin(asr_A_CS, cor_M_L ~ V5/(sqrt(V3)*sqrt(V6)))
nadiv:::pin(asr_A_CS, cor_S_L ~ V4/(sqrt(V1)*sqrt(V6)))
```

```
##          Estimate       SE
## cor_S_M -0.774119 0.186979
##          Estimate        SE
## cor_M_L 0.5717923 0.1469519
##           Estimate        SE
## cor_S_L -0.6189043 0.1912143
```

Here we see that the among-individual correlations differ greatly across combinations of 'environments': the correlation is positive between medium and large opponents, while between small:medium and small:large the correlation is negative. Looking at the raw data that we plotted at the beginning of the tutorial, this makes sense — those least aggressive against small opponents tend to be most aggressive against large opponents, and vice versa.

If P-values were desired for each of these pairiwse correlation (whether testing if $r > 0$ or against a null hypothesis of r==1; see below), we recommend using pairwise bivariate models to investigate cross-environment correlations between each pair of sub-traits.

Another useful point of character state models is that we can estimate 'repeatability' for each trait (i.e., the proportion of phenotypic variation, conditional on the fixed effects, that is explained by differences among individuals). Just as with our earlier bivariate models, we can use the `pin` function to estimate this proportion:

```
nadiv:::pin(asr_A_CS, prop_S ~ V1/(V1+V8))
nadiv:::pin(asr_A_CS, prop_M ~ V3/(V3+V10))
nadiv:::pin(asr_A_CS, prop_L ~ V6/(V6+V13))
```

```
##         Estimate         SE
## prop_S 0.3775756 0.09950306
##        Estimate         SE
## prop_M 0.432404 0.09344776
##         Estimate         SE
## prop_L 0.4350071 0.09498508
```

Here the three repeatabilities are quite similar. If you wanted to test for the significance of repeatability in aggression within this kind of model (over all opponent sizes), you could test a model fitting among-individual variation in the trivariate model (`random =~ ID:idh(trait)`) against a model without the random term, using a likelihood ratio test on 3 degrees of freedom.

Testing existence of IxE for aggression

So far we have seen that the character state approach lets us estimate among-individual variance (and repeatabilities) in, and covariance/correlation between, a set of environment-specific sub-traits. But how does this relate to IxE?

In the absence of IxE, an individual's random effect size is – by definition – constant with E. It therefore follows that the variance in random effects (i.e., the among-individual variance) is the same in each environment. It also follows that the (among-individual) correlations between environment-specific sub-traits will all be $r = +1$. So to test for IxE we should compare our trivariate model (using the US structure) to one where we set the among-individual correlations to 1 (although `ASReml` requires them to be set at 0.999):

```
init_CS_cor1_tri <- c(0.999,
                      0.999,0.999,
                      1,1,1)
names(init_CS_cor1_tri) <- c("F",
                             "F","F",
                             "U","U","U")

asr_A_CS_cor1_tri <- asreml(cbind(agg_S,
                                  agg_M,
                                  agg_L) ~ trait +
                              trait:scale(body_size) +
                              trait:block +
                              trait:opp_order,
                            random =~ ID:corgh(trait,
                                               init = init_CS_cor1_tri),
                            rcov =~ units:us(trait),
                            data = df_plast_CS,
                            maxiter = 500)
```

```
pchisq(2*(asr_A_CS$loglik - asr_A_CS_cor1_tri$loglik),
       3, lower.tail = FALSE)
```

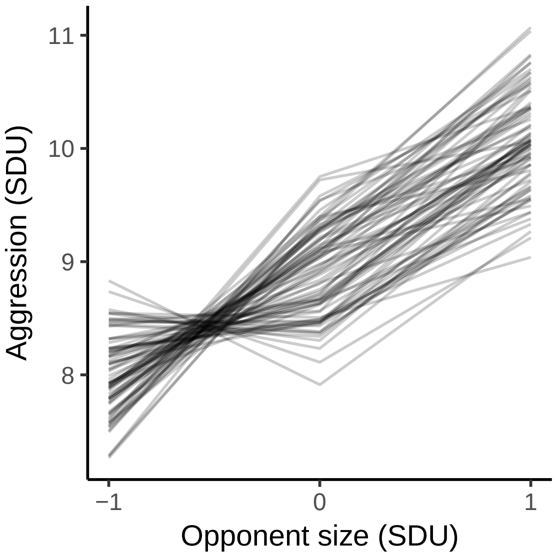
```
## [1] 1.3678e-06
```

Here we can see that the unconstrained model — in which IxE is allowed — is a (statistically) significantly better fit, so there **is** IxE overall.

Plotting the character state model

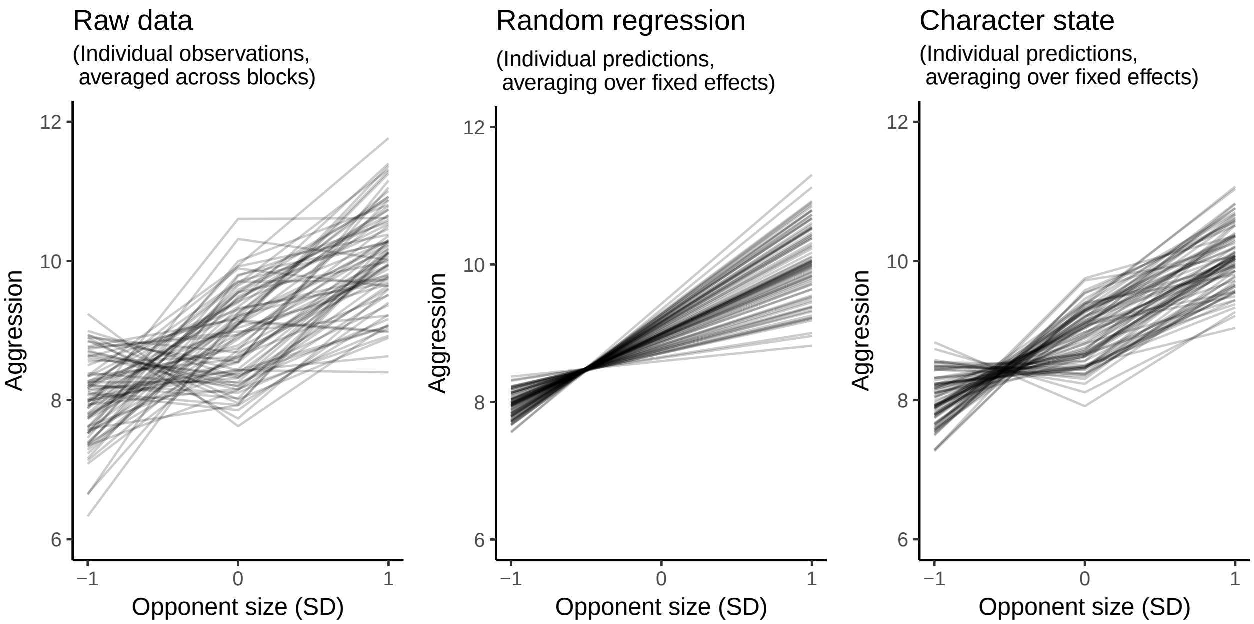
Similar to how we visualised the random regression model, we can use BLUPs to extract individual-level predictions from the model to plot predicted aggression for each individual across each environment. Here we use ASReml's `predict` function to get predicted values for each `trait` and for every `ID`, averaging over the other effects in the model. For ease of plotting, we create a new variable that we set to -1/0/1 depending on which opponent size environment the prediction is for.

```
# Retrieve individual preictions from
#  ASReml trivariate model and
#  reform into data frame for plotting
df_CS_pred <- as.data.frame(predict(asr_A_CS,
                                    classify = "trait:ID")$prediction$pvals)

# Add numeric variable for easier plotting
#  of opponent size
df_CS_pred <- df_CS_pred %>%
  mutate(sizeNum = ifelse(trait == "agg_S", -1,
                          ifelse(trait == "agg_M", 0, 1)))

ggplot(df_CS_pred, aes(x = sizeNum,
                       y = predicted.value,
                       group = ID)) +
  geom_line(alpha = 0.2) +
  scale_x_continuous(breaks = c(-1,0,1)) +
  labs(x = "Opponent size (SDU)",
       y = "Aggression (SDU)") +
  theme_classic()
```



Here, this brings together all facets of our character state analysis. We can see that (i) there is an increase in average aggression due to opponent size; (ii) correlations differ such that small:medium is negative, while medium:large is positive; (iii) there is greater among-individual variation in aggression when the opponent is large compared to when the opponent is small; (iv) there are changes in the rank order of individual aggression across environments (crossing reaction norms), so among-individual correlations are not perfect and we have evidence of IxE for aggression against different opponent sizes.

We can also compare this model fit to both the original data, and the fit from the ASReml-R random regression:



Clearly, the character state model looks like a better fit to the data than the random regression. We should note that the character state model has used more parameters than random regression, enabling this better fit — and note also that these additional parameters are not only at the among-individual level, but also at the residual/within-individual level. That is, residual variation can differ for each 'sub-trait'. There is a concern that random regression might also be anti-conservative if we assume that the residual variance is homogeneous at all values of the environmental variable (here, opponent size).

Adding fitness to character state model

As we discussed earlier, one of the challenges with reaction norm models can be that adding (and, more pertinently, interpreting associations with) another trait (or response variable) can be confusing. Given suitable data, character state models are an excellent solution because we can simply add a further response variable and fit all correlations. Instead of thinking about how selection acts on plasticity, we will be thinking about how selection on the trait (aggression) changes with environment (opponent size). Crucially (although perhaps not obviously!) **they are the same thing, just seen from alternative viewpoints** (reaction norm vs character state).

Now, let's add `fitness` to create a 4-variable response trait in a new model. Remember that `fitness` has only a single measure per individual, so we set its residual (within-individual) variance to essentially zero, and all of its residual correlations to zero accordingly.

Also note that we use the `at` keyword to apply the fixed effects only to the aggression observations (i.e., traits 1:3 of the 4-trait response).

```r
init_CS_1 <- c(0.1,
               0.1,0.1,
               0.1,0.1,0.1,
               1,1,1,1)
names(init_CS_1) <- c("U",
                      "U","U",
                      "U","U","U",
                      "U","U","U","U")
```

```
init_CS_res_1 <- c(0.1,
                   0.1,0.1,
                   0,0,0,
                   1,1,1,1e-08)
names(init_CS_res_1) <- c("U",
                          "U","U",
                          "F","F","F",
                          "U","U","U","F")

asr_A_CS_fit <- asreml(cbind(agg_S,
                             agg_M,
                             agg_L,
                             fitness) ~ trait +
                           at(trait,1:3):scale(body_size) +
                           at(trait,1:3):block +
                           at(trait,1:3):opp_order,
                         random =~ ID:corgh(trait, init = init_CS_1),
                         rcov =~ units:corgh(trait, init = init_CS_res_1),
                         data = df_plast_CS,
                         maxiter = 1000)


plot(residuals(asr_A_CS_fit) ~ fitted(asr_A_CS_fit))
qqnorm(residuals(asr_A_CS_fit))
hist(residuals(asr_A_CS_fit))
```

```
summary(asr_A_CS_fit)$varcomp[,c("component","std.error")]
```

```
##                                           component   std.error
## ID:trait!trait.agg_M:!trait.agg_S.cor   -0.77553432 0.186608955
## ID:trait!trait.agg_L:!trait.agg_S.cor   -0.62588021 0.190485210
## ID:trait!trait.agg_L:!trait.agg_M.cor    0.57307214 0.146652222
## ID:trait!trait.fitness:!trait.agg_S.cor -0.17795552 0.152280357
## ID:trait!trait.fitness:!trait.agg_M.cor  0.22224431 0.143095433
## ID:trait!trait.fitness:!trait.agg_L.cor  0.60267728 0.113522645
## ID:trait!trait.agg_S                     0.19353348 0.063294412
## ID:trait!trait.agg_M                     0.24597064 0.071046304
## ID:trait!trait.agg_L                     0.31511852 0.091788607
## ID:trait!trait.fitness                   0.02614455 0.004159899
## R!variance                               1.00000000          NA
## R!trait.agg_M:!trait.agg_S.cor           0.03321207 0.115227505
## R!trait.agg_L:!trait.agg_S.cor          -0.02243732 0.115854812
## R!trait.agg_L:!trait.agg_M.cor           0.44051567 0.093405715
## R!trait.fitness:!trait.agg_S.cor         0.00000000          NA
## R!trait.fitness:!trait.agg_M.cor         0.00000000          NA
## R!trait.fitness:!trait.agg_L.cor         0.00000000          NA
## R!trait.agg_S                            0.31783236 0.051929449
## R!trait.agg_M                            0.32252169 0.052505083
## R!trait.agg_L                            0.40659500 0.066887537
## R!trait.fitness                          0.00000001          NA
```

Again, as a sanity check we can make sure that the aggression trait variances and correlations are the same as we found in the model without fitness added. Furthermore, the residual variance of fitness is set at almost zero (line 21), and the residual correlations featuring fitness are all fixed at zero (lines 15–17).

Lines 4–6 of the summary table are what we are really interested in here — i.e., what are the among-individual correlations between fitness and aggression in the different environments? We can see from this summary that these correlations are quite different from one another. . .

Testing overall correlation structure between aggression and fitness

We can test whether there is a significant overall correlation between aggression (across all environments) and fitness by testing the unconstrained model with one in which we set all aggression:fitness correlations to 0:

```
init_CS_2 <- c(0.1,
               0.1,0.1,
               0,0,0,
               1,1,1,1)
names(init_CS_2) <- c("U",
                      "U","U",
                      "F","F","F",
                      "U","U","U","U")


asr_A_CS_fit_2 <- asreml(cbind(agg_S,
                         agg_M,
                         agg_L,
                         fitness) ~ trait +
                     at(trait,1:3):scale(body_size) +
                     at(trait,1:3):block +
                     at(trait,1:3):opp_order,
                   random =~ ID:corgh(trait, init = init_CS_2),
                   rcov =~ units:corgh(trait, init = init_CS_res_1),
                   data = df_plast_CS,
                   maxiter = 1000)
```

```
pchisq(2*(asr_A_CS_fit$loglik - asr_A_CS_fit_2$loglik),
       3, lower.tail = FALSE)
```

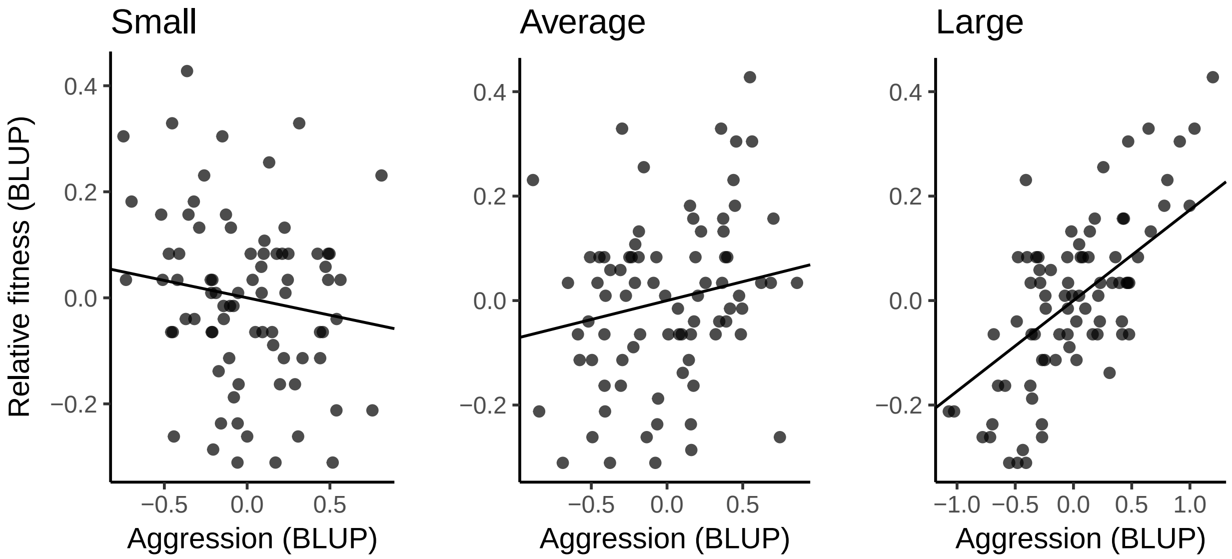
```
## [1] 0.000227296
```

So the overall association between aggression and fitness is significant. However, from the summary of the unconstrained model (previous page, `asr_A_CS_fit`) you should also be able to see that the correlations between fitness and opponent size-specific aggression vary widely across opponent size classes.

It is worth highlighting that the `pin` function can also be used to calculate fitness:subtrait covariances (using the correlations and variances of interest), which are actually environmental-specific selection differentials on the repeatable part of aggression. Here we will plot the environment-specific associations using BLUPs extracted from the unconstrained model, and use the (co)variances from the model to plot the regression line for each environment.

Below we provide the code for plotting the association between relative fitness and individual deviations (BLUPs) for aggression at the level of ‘small’ opponents — you can repeat this process for medium and large opponents, making the necessary changes:

```
# Retrieve BLUPs from ASReml trivariate model
#  and reform into data frame for plotting
df_sf_coefs <- data_frame(Trait = attr(asr_A_CS_fit$coefficients$random, "names"),
                          Value = asr_A_CS_fit$coefficients$random) %>%
  separate(Trait, c("ID","Trait"), sep = ":") %>%
  filter(Trait %in% c("trait_agg_S", "trait_fitness")) %>%
  spread(Trait, Value)

# Find the regression line for this opponent size:
#  the covariance of boldness(S), relative fitness
#   (calculated from the correlation and variances)
#  divided by the variance in boldness(S)
s_fit_slope <-  as.numeric(nadiv:::pin(asr_A_CS_fit,
                                       slope ~ (V4*sqrt(V7)*sqrt(V10))/
                                         V7)$Estimate)

gg_s_fit <- ggplot(df_sf_coefs, aes(x = trait_agg_S,
                                    y = trait_fitness,
                                    group = ID)) +
  geom_point(alpha = 0.7) +
  geom_abline(intercept = 0, slope = s_fit_slope) +
  labs(x = "Aggression (BLUP)",
       y = "Relative fitness (BLUP)",
       title = "Small") +
  theme_classic()
```



So qualitatively at least, it seems that estimated selection on individual aggressiveness goes from weakly negative with small opponents, through to more strongly positive for large opponents. If you wanted to produce a P-value for the significance of the fitness-phenotype relationship in each environment, then a simple approach would be to **fit and compare two bivariate models for each subtrait** — one in which the among-individual aggresion-fitness covariance (or correlation) is freely estimated, and one in which it is constrained to equal zero. An LRT with a single degree of freedom could be used to test whether the relationship is significantly diferent from zero.

We demonstrate this here with fitness and aggression measured against **large** opponents:

```
init_res_biv <- c(0.1, 1e-08)
names(init_res_biv) <- c("P","F")

# Model covariance
asr_A_CS_fit_lfit_us <- asreml(cbind(agg_L,
                                     fitness) ~ trait +
                                 at(trait,1):scale(body_size) +
                                 at(trait,1):block +
                                 at(trait,1):opp_order,
                               random =~ ID:us(trait,
                                               init = c(1,
                                                        0.1,1)),
                               rcov =~ units:idh(trait,
                                                 init = init_res_biv),
                               data = df_plast_CS,
                               maxiter = 1000)

# Do not model covariance
asr_A_CS_fit_lfit_idh <- asreml(cbind(agg_L,
                                     fitness) ~ trait +
                                 at(trait,1):scale(body_size) +
                                 at(trait,1):block +
                                 at(trait,1):opp_order,
                               random =~ ID:idh(trait,
                                               init = c(1,1)),
                               rcov =~ units:idh(trait,
                                                 init = init_res_biv),
                               data = df_plast_CS,
                               maxiter = 1000)
```

```
pchisq(2*(asr_A_CS_fit_lfit_us$loglik - asr_A_CS_fit_lfit_idh$loglik),
       1, lower.tail = FALSE)
```

```
## [1] 1.686584e-05
```

We find that the covariance between among-individual variation in aggression against large opponents and fitness is statistically significant. If you follow the same strategy for the other two aggression measures you should actually that neither is statistically significant:

```
# Small opponents
pchisq(2*(asr_A_CS_fit_sfit_us$loglik - asr_A_CS_fit_sfit_idh$loglik),
       1, lower.tail = FALSE)
```

```
## [1] 0.2507142
```

```
# Average-size opponents
pchisq(2*(asr_A_CS_fit_mfit_us$loglik - asr_A_CS_fit_mfit_idh$loglik),
       1, lower.tail = FALSE)
```

```
## [1] 0.1397032
```

Conclusions

Using both random regression and character state models, we find evidence of significant among-individual variance in aggressiveness. We also show that, on average, there is a plastic response manifest as higher aggression toward larger opponents. Since IxE is present we can also say that individuals differ in their plastic responses to opponent size. In our view however, the magnitude and nature of the IxE is more readily interpreted under the character state model because this enables us to estimate environment-specific among-individual variances and covariances (in addition to their derived repeatabilities and correlations). Importantly, the fact that under a null hypothesis of no IxE we expect the variances to be equal and the cross-environment correlations to be +1 means we have a clear point of comparison against which to both test and interpret the IxE.

We found strong associations between higher levels of aggression and fitness using both types of model. However, while the results of the random regression model can be interpreted in terms of selection on plasticity and intercept, there are dangers of doing so if one doesn't fully understand (i) the dependence of model outputs on the scaling/centring of the X axis, and/or (ii) the need to fully account for slope-intercept covariance, if the goal is to test for "selection on plasticity". In contrast, the character state model yields estimates of selection in each environment that are easily interpreted. Using this method, we found that the only significant covariance between fitness and aggression was when aggression was measured against large opponents. Here, greater aggression was significantly linked to higher fitness. Aggression against average and small opponents was not significantly associated with fitness.

Data wrangling

Below, we demonstrate how to do some 'data wrangling' to turn the standard long format into a different ('wide') format that can be used for character state models.

We need to do some rearranging of our data frame to structure it for this type of model. We want to end up with the aggression values for each 'environment' (for a single block) on a single row in the data frame, with each sub-trait in a different column. Each row will also have the observation of block, and the opponent body size measurement for that block.

First, we should name these traits `agg_S`, `agg_M`, and `agg_L` respectively (and we are reloading the original data first into a new data frame):

```
df_plast_2 <- read_csv("aggression.csv")
```

```
df_plast_2 <- df_plast_2 %>%
  mutate(opp_type = ifelse(opp_size == -1, "agg_S",
                          ifelse(opp_size == 0, "agg_M",
                                 "agg_L")))
```

This does mean that we need to turn our assay repeat variable into an 'order' variable, so that we can test for any systematic effects of opponent size order in the model:

```
# Create a new data frame, 'df_order',
#  which holds the 'order' of opponents for each individual in each block
df_order <- df_plast_2 %>%
  select(ID, block, assay_rep, opp_type) %>%
  arrange(ID, block, assay_rep) %>%
  separate(opp_type, into = c("tmp","opp_type")) %>%
```

```
  select(-tmp) %>%
  spread(assay_rep, opp_type) %>%
  unite(opp_order, `1`:`3`)
```

We now want to 'spread' our aggression observations across separate columns, such that each row has a value for aggression in each of the three opponent size classes (where 1 row represents an individual's observations for one block). Having spread these values across distinct columns, we then join the data frame of 'order' values to it.

Note: here we are **not** going to standardise aggression measurements to (for example) standard deviation units, but if you wanted to do that you must do it **before** spreading them across multiple rows. That would not only scale them by the overall phenotypic SD, but if you centred them as well you would retain differences in the mean values across opponent size classes (so you could still model population-level plasticity).

```
# Spread aggression values across different
#  opponent sizes
df_plast_CS <- df_plast_2 %>%
  select(ID, block, aggression, body_size, opp_type) %>%
  spread(opp_type, aggression)

# Add values for order of opponent sizes
df_plast_CS <- left_join(df_plast_CS,
                         df_order,
                         by = c("ID","block"))
```

. . . and quickly check to make sure the new data frame looks the way we wanted:

```
head(df_plast_CS)
```

We did not include **fitness** here because it makes the spreading of our aggression columns difficult, but we can add this back to our data frame now — remembering to make sure it is only repeated once per individual:

```
# Create new df containing non-NA fitness values for each individual:
df_fit <- df_plast_2 %>%
  select(ID, fitness) %>%
  filter(!is.na(fitness))

# Join to the CS data frame
df_plast_CS <- left_join(df_plast_CS,
                         df_fit,
                          by = "ID")

# Delete duplicates in fitness
# - easy way here is to keep only those with one of the blocks
df_plast_CS <- df_plast_CS %>%
  mutate(fitness = ifelse(block == -0.5, fitness, NA))
```

Finally, let's standardise fitness to 'relative fitness' by dividing by the population mean:

```
df_plast_CS <- df_plast_CS %>%
  mutate(fitness = fitness/mean(fitness, na.rm=TRUE))
```

. . . and check the results of our wrangling efforts:

```
head(df_plast_CS)
```

Again, http://r4ds.had.co.nz/ is the place to go to learn more about the `tidyverse` packages used for this kind of ‘data wrangling’.