Lathyrus ms2: Selection on reaction norms - multivariate modeling for phenotypic selection on plasticity 5 (Arnold et al. 2019 Phil. Trans. R. Soc. B)

Contents

Data preparation

```
# Read data
alldata<-read.csv(
  "C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms1/data/clean/alldata.csv",
  header=T,sep="\t",dec=".")
data_sel<-read.csv(
  "C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms1/data/clean/data_sel.csv",
                       header=T,sep="\t",dec=",")

# Create a subset of data with plants with 5 or more years of data
data_5yrs<-select(alldata,c(1,3,10,12,14,19))%>% # Select columns needed by now
```

```r
  right_join(unique(data_sel[c(1,171:172,183)]))
subset(subset(data_5yrs,is.na(n_intact_seeds)),!is.na(FFD))
```

```
## [1] year            id              FFD             n_fl            shoot_vol
## [6] n_intact_seeds mean_4          mean_5          min_4
## <0 rows> (or 0-length row.names)
```

```r
# 0 cases with no fitness data and FFD data
data_5yrs<-subset(data_5yrs,!is.na(n_intact_seeds)) # Select cases with fitness data
data_5yrs$id<-droplevels(data_5yrs$id)

subset(data_5yrs,is.na(FFD)&is.na(n_fl)&n_intact_seeds==0)
```

```
##       year      id FFD n_fl shoot_vol n_intact_seeds   mean_4    mean_5
## 10171 2006   new_7  NA   NA 3125.0800              0 4.611667 10.466129
## 11515 2011  new_91  NA   NA 7777.7000              0 8.438333 10.925806
## 12686 2015 new_249  NA   NA  740.5700              0 8.037472  9.777419
## 12901 2017  new_35  NA   NA  870.4407              0 4.805000 10.896774
## 12979 2017 new_216  NA   NA 1002.7558              0 4.805000 10.896774
## 13033 2017 new_408  NA   NA 2454.0461              0 4.805000 10.896774
##           min_4
## 10171 1.1233333
## 11515 3.7383333
## 12686 3.2662885
## 12901 0.5766667
## 12979 0.5766667
## 13033 0.5766667
```

```r
# These probably did not flower - remove

data_5yrs<-anti_join(data_5yrs,subset(data_5yrs,is.na(FFD)&is.na(n_fl)&n_intact_seeds==0))

subset(data_5yrs,is.na(FFD)&is.na(n_fl)&is.na(n_intact_seeds))
```

```
## [1] year            id              FFD             n_fl            shoot_vol
## [6] n_intact_seeds mean_4          mean_5          min_4
## <0 rows> (or 0-length row.names)
```

```r
# All plants/years are flowering events here

# n_years_fl = Number of years when the plant has flowered
# n_years_FFD = Number of years when we have data for FFD
# n_years_life = Number of years when the plant was alive --> Calculated by hand from data

n_years_FFD<-data_5yrs %>% group_by(id) %>% summarise(n_years_FFD = sum(!is.na(FFD)))
n_years_life<-read.table(
  "C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/data/n_years_life.txt",
  header=T)
data_5yrs<-data_5yrs%>%
 rbind(list(2016,"new_121",NA,2,NA,NA,5.665,12.20806,1.908333))
# Add data for id new_121 -> 2 flowers in 2016 (from comment in Excel files)
```

```
data_5yrs<-data_5yrs%>%
  right_join(n_years_FFD)%>%
  group_by(id)%>%
  mutate(n_years_fl=n())%>%
  arrange(.,id)%>%
  filter(n_years_FFD>=5)%>% # Keep individuals for which we have data on FFD for 5+ years
  filter(!is.na(FFD))%>% # Keep records for which we have FFD values
  droplevels()%>%
  right_join(n_years_life) # Add info on n_years_life

length(levels(data_5yrs$id)) # 163 (before 156) plant individuals
```

```
## [1] 163
```

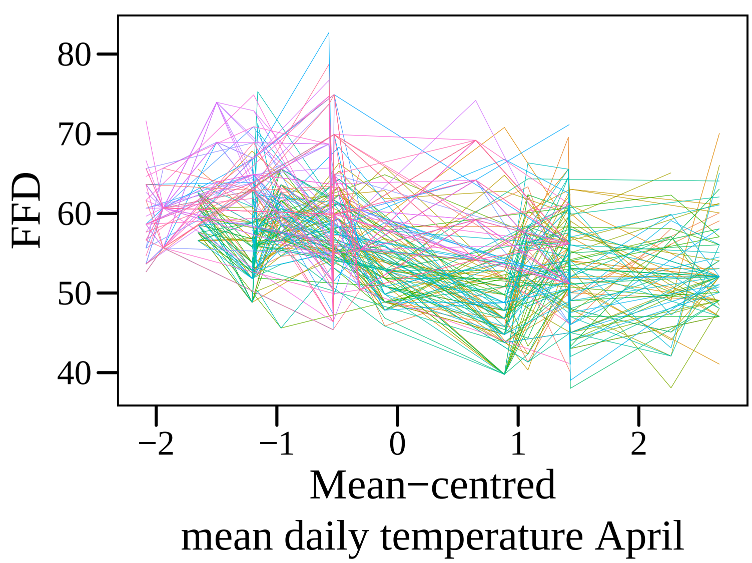
Check for non-linearities

Code based on Arnold et al 2019 New Phyt.

The x variable (mean daily temperature April) is mean-centered (substracting the mean), so the intercepts reflect average values for the population and individuals. From here on, we use this mean-centred temperature (cmean_4).

```
data_5yrs$cmean_4<-scale(data_5yrs$mean_4,center=T,scale=F)
```

Plot the main effects (raw values of FFD against mean-centred temperatures for each plant id)



Basic linear model

Fit a linear model for the fixed effect of temperature on FFD and observe the average population-level reaction norm. Note that we also add a random effect for 'year' to take account of the repeated measures at each temperature (to account for differences among the years across which each id is represented). The mixed model is to be fitted using ML rather than REML so that models that contain different fixed effects can be compared directly.

Using blmer which does maximum a posteriori estimation for linear and generalized linear mixed-effects models in a Bayesian setting. Allows the user to do Bayesian inference or penalized maximum likelihood, with priors imposed on the different model components.

```
model1.1 <- blmer(FFD ~ cmean_4 + (1|year), REML = FALSE, data = data_5yrs,
                  lmerControl(optimizer = "Nelder_Mead"))
summary(model1.1)
```

```
## Cov prior  : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 0.0012
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year)
##    Data: data_5yrs
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   7062.4   7082.7  -3527.2   7054.4     1158
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8073 -0.6319 -0.0945  0.5558  4.6521
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 23.53    4.851
##  Residual             23.55    4.853
## Number of obs: 1162, groups:  year, 22
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  57.3161     1.0496  54.610
## cmean_4      -2.4062     0.7767  -3.098
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 0.078
```

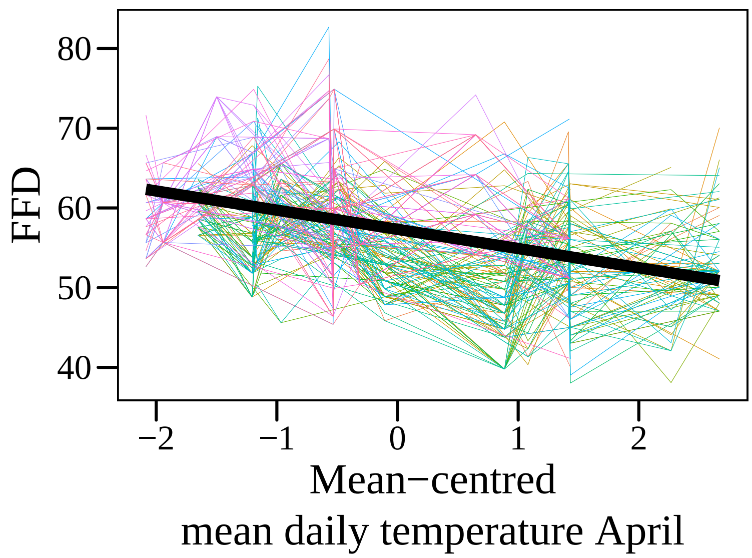
```
r.squaredGLMM(model1.1)
```

```
##            R2m       R2c
## [1,] 0.1818137 0.5907455
```

Visually assess how well the linear model fits the raw data by overlaying the regression line from model1.1 as an average population-level reaction norm. Use the predict function to predict y-values accross the continuous x-axis and then plot the fixed effect of temperature from model1.1 over the raw id-specific reaction norms.

```
temperature_pred <- data.frame(cmean_4 = seq(from = min(data_5yrs$cmean_4),
                                             to = max(data_5yrs$cmean_4),
                                             length.out = 50))
```

```
temperature_pred$fit1.1 <- predict(model1.1,
                                   newdata = temperature_pred, re.form = NA)
# re.form=NA includes no random effects

# Plot the raw data and overlay the fit of Model1.1
ggplot(temperature_pred, aes(x = cmean_4, y = fit1.1)) +
  geom_line(data = data_5yrs, aes(y = FFD, colour = id),size = 0.1) +
  geom_line(size = 2) +  ylab("FFD") +
  xlab("Mean-centred\nmean daily temperature April")+my_theme()
```



Quadratic fixed effects model

Fit a quadratic model for the fixed effect of temperature on FFD.

```
model1.2 <- blmer(FFD ~ poly(cmean_4, 2, raw = T) + (1|year),
                  REML = FALSE, data = data_5yrs,
                  lmerControl(optimizer = "Nelder_Mead"))
summary(model1.2)
```

```
## Cov prior  : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 0.0014
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ poly(cmean_4, 2, raw = T) + (1 | year)
##    Data: data_5yrs
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   7064.4   7089.7  -3527.2   7054.4     1157
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8078 -0.6318 -0.0944  0.5559  4.6522
```

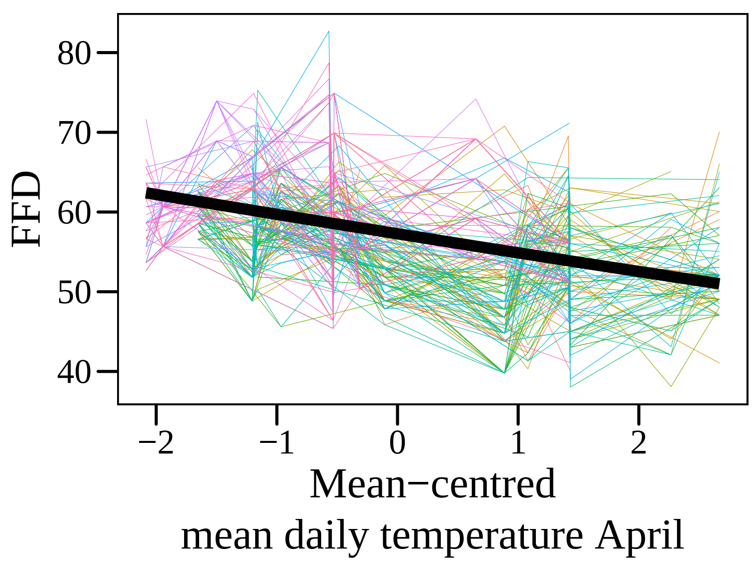
```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 23.53    4.851
##  Residual             23.55    4.853
## Number of obs: 1162, groups:  year, 22
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                57.27117    1.54472  37.076
## poly(cmean_4, 2, raw = T)1 -2.41539    0.81007  -2.982
## poly(cmean_4, 2, raw = T)2  0.02409    0.60728   0.040
##
## Correlation of Fixed Effects:
##              (Intr) p(_4,2,r=T)1
## p(_4,2,r=T)1  0.260
## p(_4,2,r=T)2 -0.734 -0.284
```

```
r.squaredGLMM(model1.2)
```

```
##            R2m       R2c
## [1,] 0.1815884 0.5906088
```

Predict values based on the model fit and plot the overall model fit over the top of the raw data.

```
temperature_pred$fit1.2 <- predict(model1.2,
                                   newdata = temperature_pred, re.form = NA)

ggplot(temperature_pred, aes(x = cmean_4, y = fit1.2)) +
  geom_line(data = data_5yrs, aes(y = FFD, colour = id),size = 0.1) +
  geom_line(size = 2) + ylab("FFD") +
  xlab("Mean-centred\nmean daily temperature April")+my_theme()
```



Compare with previous model using likelihood ratio test and AIC.

```r
chi2 <- 2*(summary(model1.2)$logLik - summary(model1.1)$logLik)
1-pchisq(chi2,1)
```

```
## 'log Lik.' 0.9666321 (df=5)
```

```r
AIC(model1.1, model1.2)
```

```
##          df      AIC
## model1.1  4 7062.434
## model1.2  5 7064.432
```

Lower R2, non-significant LRT p-value and larger AIC. Model1.1 with only linear (non-quadratic effects) is better.

Linear fixed effects with random intercepts model

Fit a linear mixed effects model (random intercepts only) for the fixed effect of temperature on FFD and random effect of id intercepts. We are allowing the y-intercept value to vary among ids.

```r
model1.3 <- blmer(FFD ~ cmean_4 + (1|year) + (1|id),
                  REML = FALSE, data = data_5yrs,
                  lmerControl(optimizer = "Nelder_Mead"))
summary(model1.3)
```

```
## Cov prior  : id ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 2.45
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 | id)
##    Data: data_5yrs
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   7015.6   7040.9  -3502.8   7005.6     1157
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0280 -0.6266 -0.0884  0.5610  4.4828
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  3.446   1.856
##  year     (Intercept) 23.160   4.812
##  Residual             20.215   4.496
## Number of obs: 1162, groups:  id, 163; year, 22
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  57.3981     1.0503  54.648
```

```
## cmean_4      -2.3996     0.7707  -3.114
##
## Correlation of Fixed Effects:
##        (Intr)
## cmean_4 0.077
```

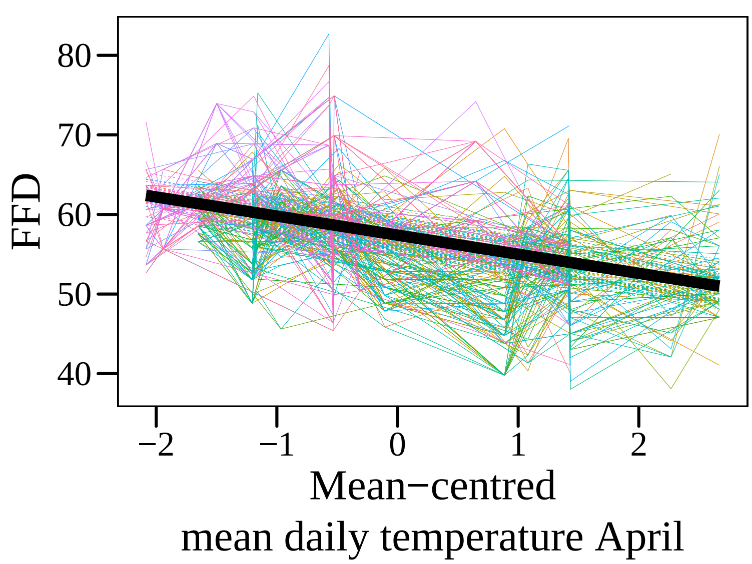
```
r.squaredGLMM(model1.3)
```

```
##           R2m       R2c
## [1,] 0.181835 0.6467525
```

The outcome of the mixed-effects model is the linear effect of temperature on FFD, whilst allowing the intercepts of each id's FFD to account for some of the residual variance in the model.

Predict values based on the model fit and plot the overall model fit over the top of the raw data.

```
temperature_pred$fit1.3 <- predict(model1.3,
                                   newdata = temperature_pred, re.form = NA)

# Make a prediction for the average population-level mean reaction norm
# and append it to the dataset
data_5yrs$pred_pop1.3  <- predict(model1.3, re.form = NA)
# Make predictions for each id-level reaction norm
data_5yrs$pred_id1.3 <- predict(model1.3, re.form = ~(1|id))

# Plot predicted id reaction norms over the raw data, along with the overall mean
ggplot(temperature_pred, aes(x = cmean_4, y = fit1.3)) +
  geom_line(data = data_5yrs, aes(y = pred_id1.3, group = id, colour = id),
            lty = 2,size = 0.1)+
  geom_line(data = data_5yrs, aes(y = FFD, group = id, colour = id),size = 0.1) +
  geom_line(size = 2) +
  ylab("FFD") + xlab("Mean-centred\nmean daily temperature April") +  my_theme()
```



Compare with previous model using likelihood ratio test and AIC.

```r
chi2 <- 2*(summary(model1.3)$logLik - summary(model1.1)$logLik)
1-pchisq(chi2, 1)
```

```
## 'log Lik.' 2.81497e-12 (df=5)
```

```r
AIC(model1.1, model1.2, model1.3)
```

```
##          df      AIC
## model1.1  4 7062.434
## model1.2  5 7064.432
## model1.3  5 7015.621
```

R2 has not substantially increased, but significant LRT p-value and smaller AIC. Model1.3 with random intercepts explains more residual variance than model1.1 without trading-off against increased model complexity.

This means that there is significant variation in intercepts of the RN among individuals.

Linear fixed effects with linear random regression model

Fit a linear mixed effects model for the fixed effect of temperature on FFD and random effect of id intercepts and slopes. Allows the slopes of ids to vary in addition to the intercepts, so that the random regression slopes might be fit better to the observed patterns in the raw data. The addition of '+x-variable' (here: '1+cmean_4') to the left side of the random effect term ('|id') in model1.3 allows the slopes of the random id regressions to vary across mean-centred temperature.

We use blmer instead of lmer because otherwise this model has a singular fit.

```r
model1.4 <- blmer(FFD ~ cmean_4 + (1|year) + (1+cmean_4|id), REML = FALSE,
                  data = data_5yrs,lmerControl(optimizer = "Nelder_Mead"))
summary(model1.4)
```

```
## Cov prior  : id ~ wishart(df = 4.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 9.015
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 + cmean_4 | id)
##    Data: data_5yrs
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   6995.7   7031.1  -3490.8   6981.7     1155
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0659 -0.6121 -0.0772  0.5855  4.1683
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept)  3.3119  1.8199
```

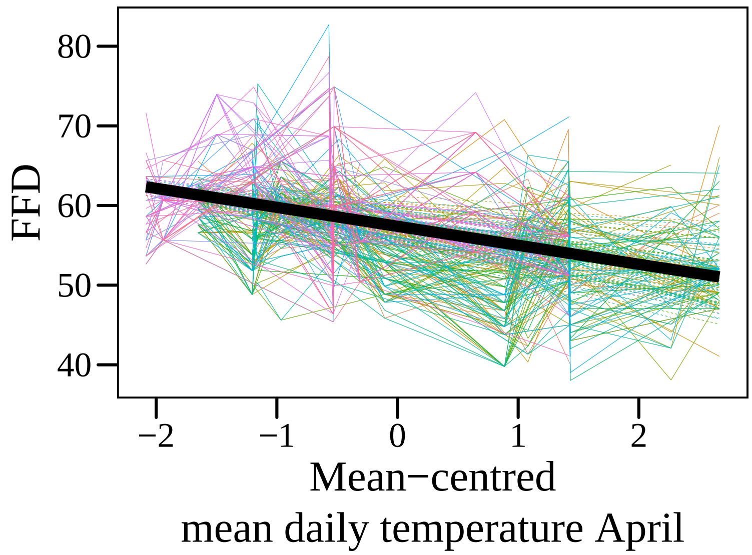
```
##           cmean_4      0.6341  0.7963   0.80
##  year     (Intercept) 23.3504  4.8322
##  Residual             19.2167  4.3837
## Number of obs: 1162, groups:  id, 163; year, 22
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  57.3882     1.0538  54.458
## cmean_4      -2.3818     0.7763  -3.068
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 0.088
```

```r
r.squaredGLMM(model1.4)
```

```
##            R2m       R2c
## [1,] 0.1789953 0.6644894
```

Predict values based on the model fit and plot the overall model fit over the top of the raw data.

```r
temperature_pred$fit1.4 <- predict(model1.4, newdata = temperature_pred, re.form = NA)

# Make a prediction for the population-level mean reaction norm
# and append it to the dataset
data_5yrs$pred_pop1.4  <- predict(model1.4, re.form = NA)
# Make predictions for the id-level reaction norms
data_5yrs$pred_id1.4 <- predict(model1.4, re.form = ~(1+cmean_4|id))

# Plot predicted id reaction norms over the raw data, along with the overall mean
ggplot(temperature_pred, aes(x = cmean_4, y = fit1.4)) +
  geom_line(data = data_5yrs, aes(y = pred_id1.4, group = id, colour = id),
            lty = 2,size = 0.1) +
  geom_line(data = data_5yrs, aes(y = FFD, group = id, colour = id),size = 0.1) +
  geom_line(size = 2) +
  ylab("FFD") + xlab("Mean-centred\nmean daily temperature April") +  my_theme()
```



Compare with previous model using likelihood ratio test and AIC.

```
chi2 <- 2*(summary(model1.4)$logLik - summary(model1.3)$logLik)
# The df difference between models can be checked by
# looking at the df within the models being compared
summary(model1.3)$logLik
```

```
## 'log Lik.' -3502.81 (df=5)
```

```
summary(model1.4)$logLik
```

```
## 'log Lik.' -3490.831 (df=7)
```

```
# Note that between model1.3 and model1.4 there is a change of 2 df, so the
# pchisq change needs to be specified with 2 df rather than 1 as in previous comparisons.
1-pchisq(chi2, 2)
```

```
## 'log Lik.' 6.273829e-06 (df=7)
```

```
AIC(model1.1, model1.2, model1.3, model1.4)
```

```
##          df      AIC
## model1.1  4 7062.434
## model1.2  5 7064.432
## model1.3  5 7015.621
## model1.4  7 6995.662
```

Significant LRT p-value and smaller AIC. The random regression mixed model (model1.4) has significantly improved the model fit to the data.

This means that there is also significant variation in slopes of the RN among individuals.

Linear fixed effects with quadratic random regression model

A final attempt to improve the model fit is to allow the random effect of id to vary in not only intercept and slope, but also in curvature, by fitting an additional quadratic random effect term.

Fit a linear mixed effects model for the fixed effect of growth temperature on FFD and random effect of id intercepts, slopes, and curvature

```
model1.5 <- blmer(FFD ~ cmean_4 + (1|year) + (1 + cmean_4 + I(cmean_4^2)|id),
                  REML = FALSE,data = data_5yrs,lmerControl(optimizer = "Nelder_Mead"))
summary(model1.5)
```

```
## Cov prior  : id ~ wishart(df = 5.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 18.1613
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 + cmean_4 + I(cmean_4^2) | id)
##    Data: data_5yrs
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   7002.6   7053.1  -3491.3   6982.6     1152
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1066 -0.6180 -0.0770  0.5896  4.1526
##
## Random effects:
##  Groups   Name         Variance Std.Dev. Corr
##  id       (Intercept)   3.92903 1.9822
##           cmean_4       0.67532 0.8218    0.79
##           I(cmean_4^2)  0.04727 0.2174   -0.50 -0.22
##  year     (Intercept)  23.31294 4.8283
##  Residual              19.07219 4.3672
## Number of obs: 1162, groups:  id, 163; year, 22
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  57.3863     1.0529  54.504
## cmean_4      -2.3818     0.7756  -3.071
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 0.088
```

```
r.squaredGLMM(model1.5)
```

```
##            R2m       R2c
## [1,] 0.1790645 0.6668703
```

Compare with previous model using likelihood ratio test and AIC.

```r
chi2 <- 2*(summary(model1.5)$logLik - summary(model1.4)$logLik)
# The df difference between models can be checked by
# looking at the df within the models being compared
summary(model1.4)$logLik
```

```
## 'log Lik.' -3490.831 (df=7)
```

```r
summary(model1.5)$logLik
```

```
## 'log Lik.' -3491.279 (df=10)
```

```r
# Note that between model1.3 and model1.4 there is a change of 3 df, so the pchisq
# change needs to be specified with 3 df rather than 1 or 2 as in previous comparisons.
1-pchisq(chi2, 3)
```

```
## 'log Lik.' 1 (df=10)
```

```r
AIC(model1.1, model1.2, model1.3, model1.4, model1.5)
```

```
##          df      AIC
## model1.1  4 7062.434
## model1.2  5 7064.432
## model1.3  5 7015.621
## model1.4  7 6995.662
## model1.5 10 7002.558
```

Model1.4 is the best model for these data. We will proceed hereafter with model1.4 to extract the best linear unbiased predictors (BLUPs) for each id.

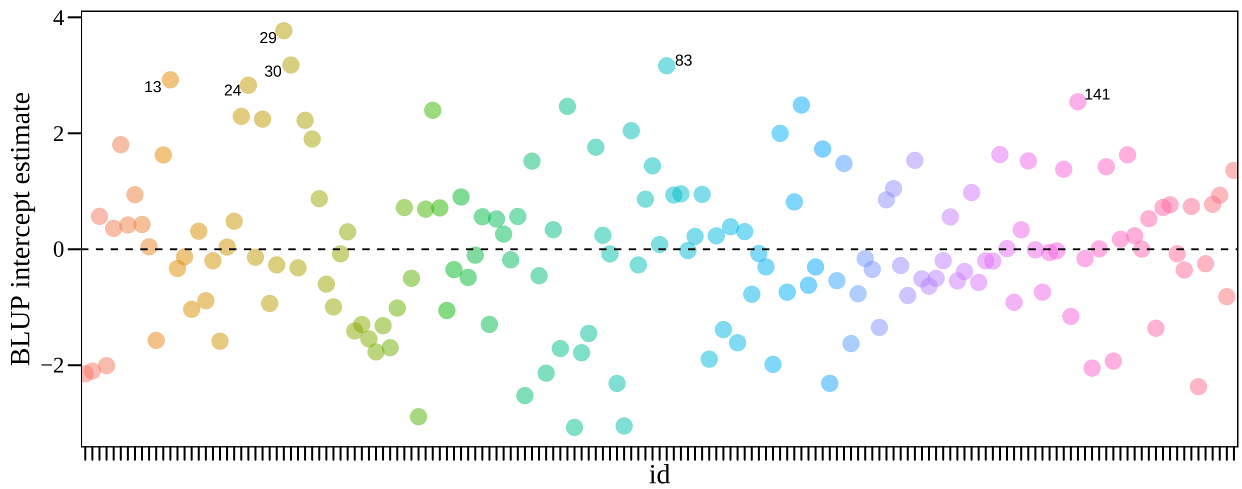
Extract BLUPs from model1.4 (linear random regression mixed model)

BLUPs represent the response of a given id to the fixed effect of temperature as the difference between that id's predicted response and the population-level average predicted response. Here, we calculate and plot BLUPs for ranking plasticity.

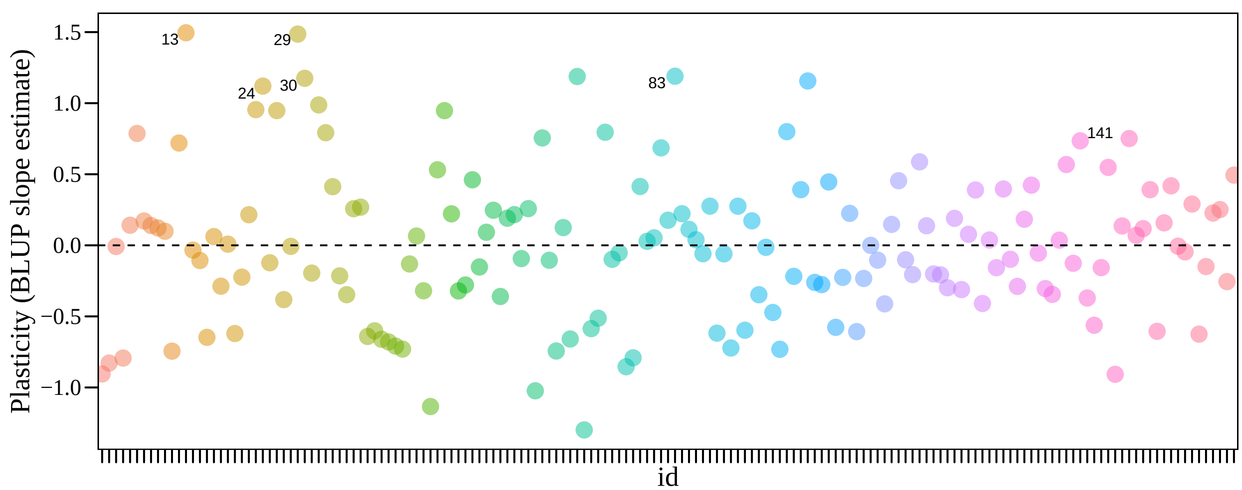
```r
id_blups <- ranef(model1.4)$`id`
id_index <- as.factor(c(1:163))
id_data  <- cbind(id_index, id_blups)
colnames(id_data) <- c("id", "BLUP_int", "BLUP_slope")
with(id_data,cor(BLUP_int,BLUP_slope)) # highly correlated!
```

```
## [1] 0.9620133
```

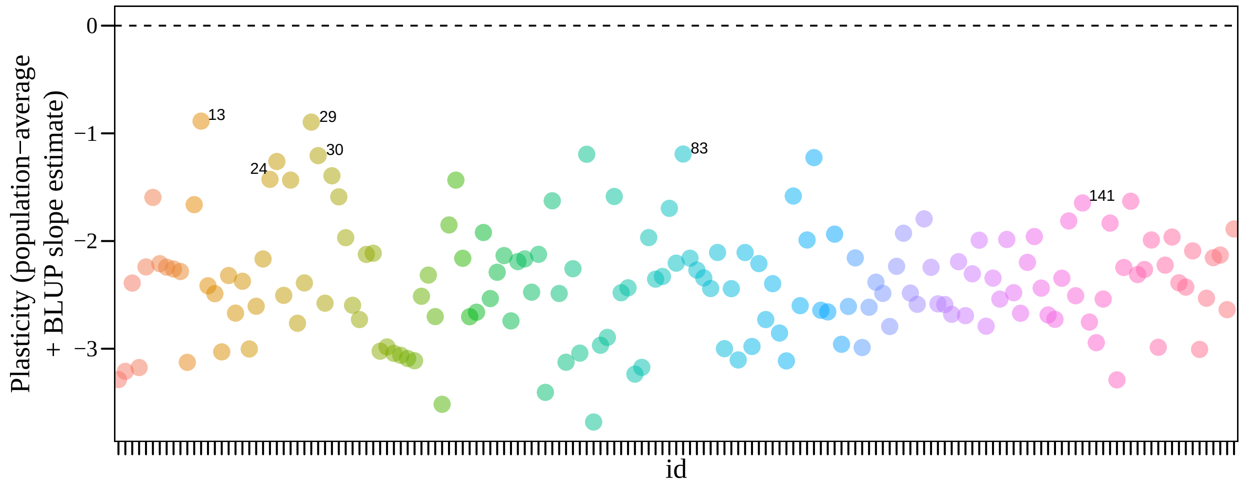
The BLUP intercept term indicates the difference in id elevation relative to the population-average, so more positive values of BLUP intercept indicate that the id's reaction norm occurs above the population-level average and negative values are below the population-level average. The BLUP intercept values are not a measure of plasticity, but these values may be correlated with BLUP slope values and otherwise may be a parameter of interest for comparing among ids.



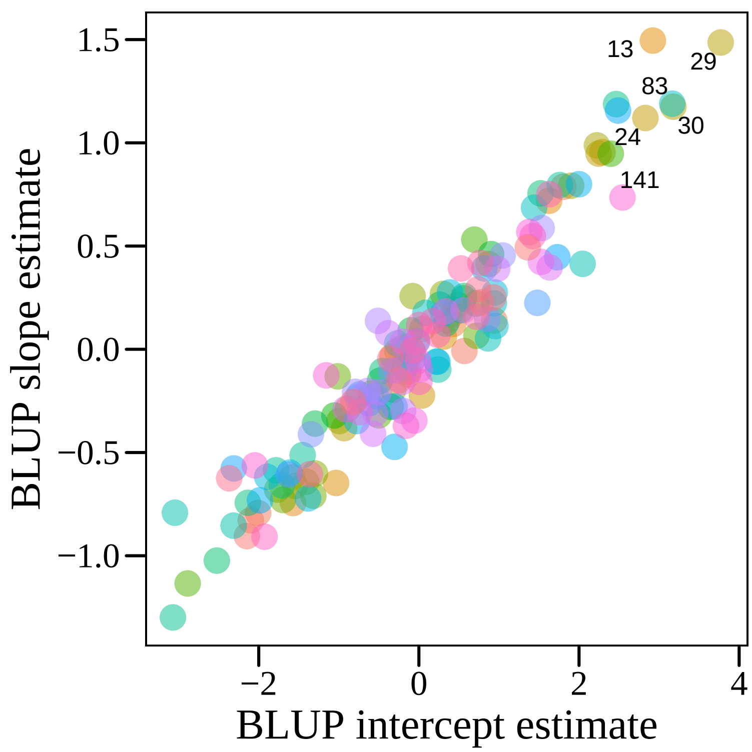
The BLUP slope estimate is the difference in slope (relative steepness of change) between the population-level average response and the response of the id. Here, that is the difference in slope of FFD for each value of temperature relative to the population-level average slope.



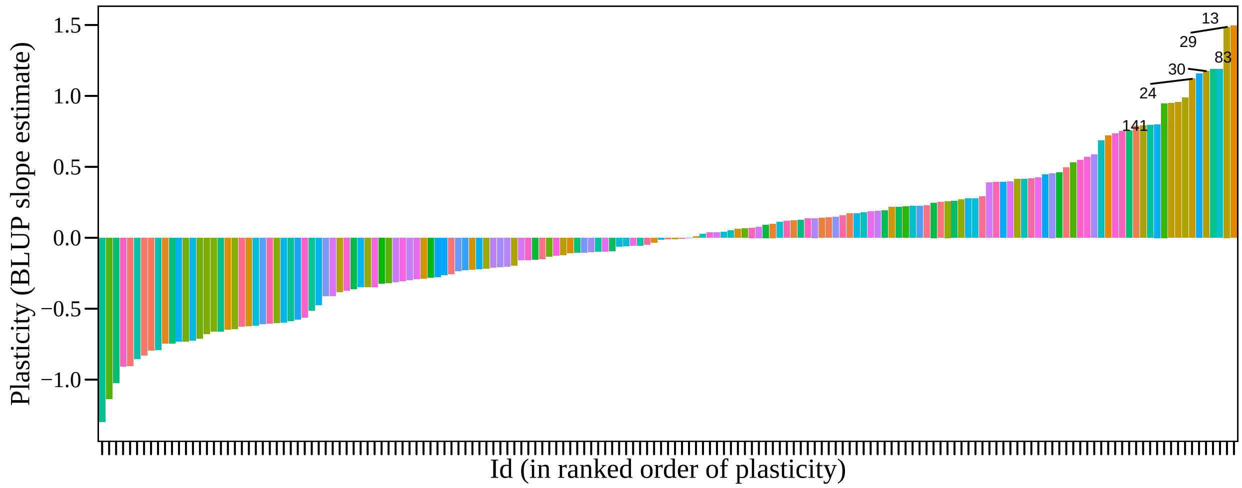
We now add the BLUP slopes for the ids to the population average. Because the population-level average response is negative overall, all ids have a negative slope when the BLUP slope estimates are added to the population-level average slope estimate from model1.4.



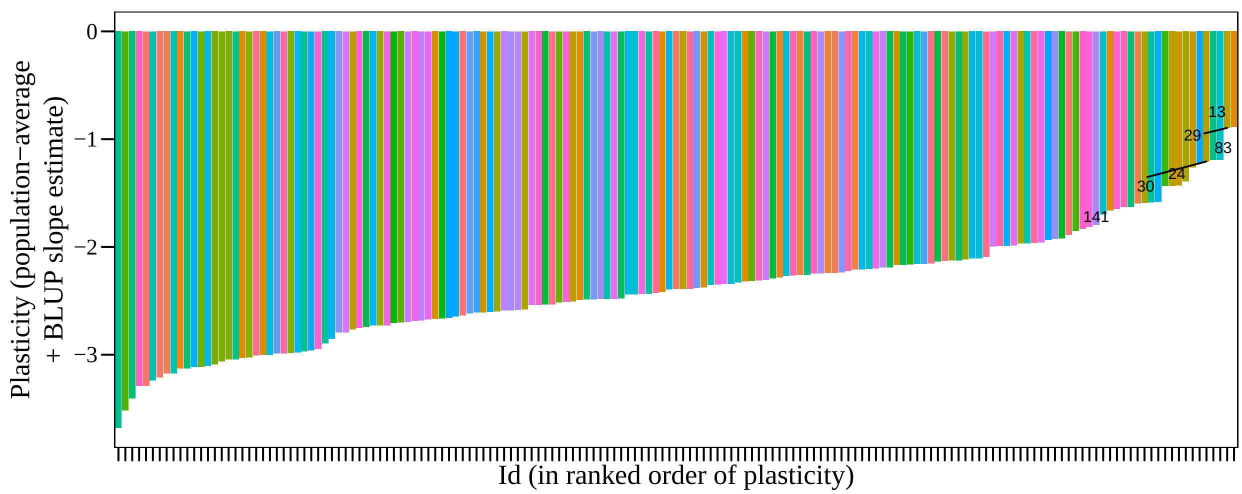
The BLUP intercept and slope estimates are sometimes correlated. The correlation coefficient is given in the random effects correlation from the model1.4 summary, which is 0.80. This positive relationship can clearly be seen when plotting the BLUP slope estimate against the BLUP intercept estimate. Ids with the most positive BLUP slope estimate (labelled) have the highest positive intercept, and have the least plasticity across growth temperatures (see previous figs with the same individuals labelled).



We can rank the BLUPs in order: sorting BLUPs by slope of most to least plastic. Because the population-level average response is negative, the most negative BLUP slope estimates represent steeper reaction norm slopes and hence greater plasticity, and more positive BLUP slope estimates represent flatter reaction norms and less plasticity in FFD in response to temperatures.



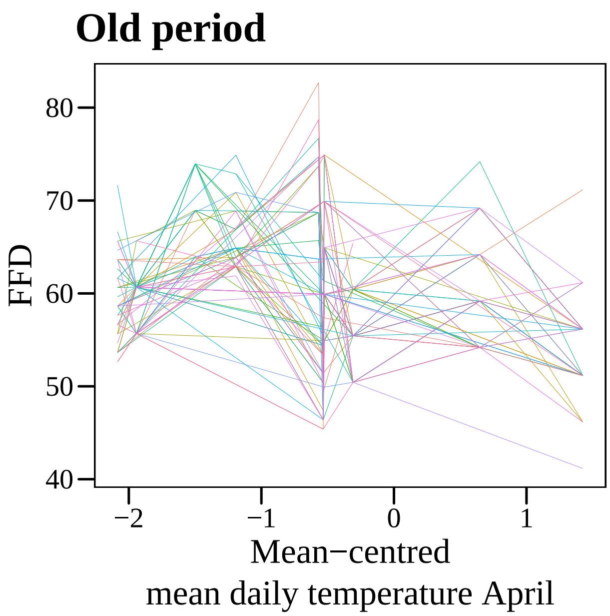
Another way to visualise the plasticity rank for negative data is by adding the BLUP slope values to the population-level average effect of temperature.

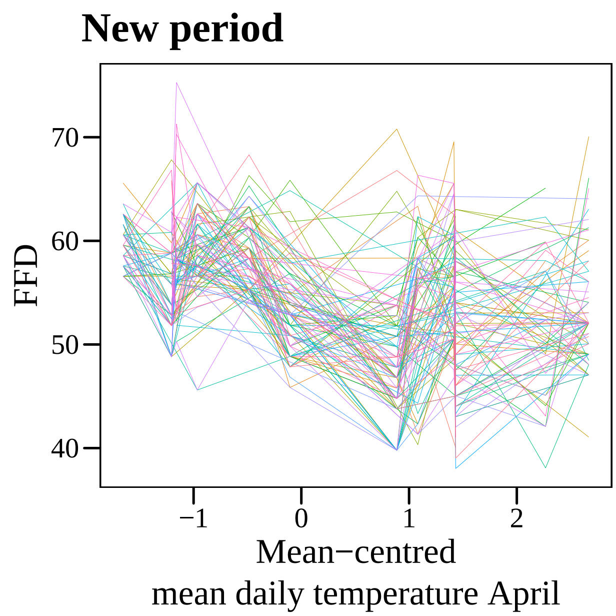


Important note! BLUPs estimated from linear mixed-effects regression models fitted in lme4 are single point estimates that do not have associated measures of uncertainty. As a result, any derived statistics or formal interpretation of plasticity based on these BLUPs is potentially very dangerous and anticonservative without properly accounting for estimation uncertainty. For using BLUPs beyond simple ranking (e.g., of the least to most plastic genotypes), it is strongly encouraged to read the references provided here to avoid the misuse of BLUPs by using a Bayesian MCMC framework (e.g., by using the MCMCglmm package in R) to generate estimates of uncertainty around BLUPs (Hadfield, 2010; Hadfield et al., 2010; Houslay & Wilson, 2017).

Check for non-linearities (two periods)

Plot the main effects (raw values of FFD against mean-centred temperatures for each plant id)





Basic linear model

```
model1.1old <- blmer(FFD ~ cmean_4 + (1|year), REML = FALSE,
                     data = subset(data_5yrs,period=="old"),
                     lmerControl(optimizer = "Nelder_Mead"))
model1.1new <- blmer(FFD ~ cmean_4 + (1|year), REML = FALSE,
                     data = subset(data_5yrs,period=="new"),
                     lmerControl(optimizer = "Nelder_Mead"))
summary(model1.1old)
```

```
## Cov prior  : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : -0.8797
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year)
##    Data: subset(data_5yrs, period == "old")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   2361.1   2376.9  -1176.5   2353.1      388
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8450 -0.6774 -0.1196  0.5139  3.7605
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 38.26    6.185
##  Residual             21.28    4.613
## Number of obs: 392, groups:  year, 10
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   59.710      2.329  25.633
## cmean_4       -2.036      1.888  -1.079
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 0.532
```

```r
summary(model1.1new)
```

```
## Cov prior  : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 1.8589
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year)
##    Data: subset(data_5yrs, period == "new")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   4695.7   4714.3  -2343.8   4687.7      766
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7382 -0.6672 -0.0690  0.5846  4.5179
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept)  7.135   2.671
##  Residual             24.638   4.964
## Number of obs: 770, groups:  year, 12
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  55.3173     0.8166  67.740
## cmean_4      -1.6897     0.5684  -2.973
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 -0.236
```

```r
r.squaredGLMM(model1.1old)
```

```
##             R2m       R2c
## [1,] 0.07667432 0.6699549
```

```r
r.squaredGLMM(model1.1new)
```

```
##            R2m      R2c
## [1,] 0.1432801 0.335667
```

Quadratic fixed effects model

```
model1.2old <- blmer(FFD ~ poly(cmean_4, 2, raw = T) + (1|year), REML = FALSE,
                     data = subset(data_5yrs,period=="old"),
                     lmerControl(optimizer = "Nelder_Mead"))
model1.2new <- blmer(FFD ~ poly(cmean_4, 2, raw = T) + (1|year), REML = FALSE,
                     data = subset(data_5yrs,period=="new"),
                     lmerControl(optimizer = "Nelder_Mead"))
summary(model1.2old)
```

```
## Cov prior  : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : -0.749
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ poly(cmean_4, 2, raw = T) + (1 | year)
##    Data: subset(data_5yrs, period == "old")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   2362.2   2382.0  -1176.1   2352.2      387
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8483 -0.6807 -0.1210  0.5181  3.7686
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 35.06    5.922
##  Residual             21.28    4.613
## Number of obs: 392, groups:  year, 10
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                  61.064      2.720  22.450
## poly(cmean_4, 2, raw = T)1   -3.138      2.206  -1.422
## poly(cmean_4, 2, raw = T)2   -1.367      1.566  -0.873
##
## Correlation of Fixed Effects:
##              (Intr) p(_4,2,r=T)1
## p(_4,2,r=T)1  0.030
## p(_4,2,r=T)2 -0.572  0.573
```

```
summary(model1.2new)
```

```
## Cov prior  : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 2.2072
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ poly(cmean_4, 2, raw = T) + (1 | year)
##    Data: subset(data_5yrs, period == "new")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
```

```
##      AIC      BIC   logLik deviance df.resid
##   4695.1   4718.3  -2342.5   4685.1      765
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7671 -0.6581 -0.0537  0.5726  4.5196
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept)  5.656   2.378
##  Residual             24.637   4.964
## Number of obs: 770, groups:  year, 12
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 54.0266     1.0877  49.669
## poly(cmean_4, 2, raw = T)1  -2.3329     0.6492  -3.593
## poly(cmean_4, 2, raw = T)2   0.7305     0.4561   1.602
##
## Correlation of Fixed Effects:
##                (Intr) p(_4,2,r=T)1
## p(_4,2,r=T)1  0.334
## p(_4,2,r=T)2 -0.740 -0.619
```

```r
r.squaredGLMM(model1.2old)
```

```
##            R2m       R2c
## [1,] 0.1217673 0.6682877
```

```r
r.squaredGLMM(model1.2new)
```

```
##            R2m       R2c
## [1,] 0.1792669 0.3325146
```

Compare with previous model using likelihood ratio test and AIC.

```r
chi2 <- 2*(summary(model1.2old)$logLik - summary(model1.1old)$logLik)
1-pchisq(chi2,1)
```

```
## 'log Lik.' 0.3535535 (df=5)
```

```r
chi2 <- 2*(summary(model1.2new)$logLik - summary(model1.1new)$logLik)
1-pchisq(chi2,1)
```

```
## 'log Lik.' 0.1043035 (df=5)
```

```r
AIC(model1.1old, model1.2old)
```

```
##             df      AIC
## model1.1old  4 2361.054
## model1.2old  5 2362.193
```

```r
AIC(model1.1new, model1.2new)
```

```
##              df      AIC
## model1.1new  4 4695.696
## model1.2new  5 4695.058
```

Linear fixed effects with random intercepts model

```r
model1.3old <- blmer(FFD ~ cmean_4 + (1|year) + (1|id), REML = FALSE,
                     data = subset(data_5yrs,period=="old"),
                     lmerControl(optimizer = "Nelder_Mead"))
model1.3new <- blmer(FFD ~ cmean_4 + (1|year) + (1|id), REML = FALSE,
                     data = subset(data_5yrs,period=="new"),
                     lmerControl(optimizer = "Nelder_Mead"))
summary(model1.3old)
```

```
## Cov prior  : id ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 2.4505
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 | id)
##    Data: subset(data_5yrs, period == "old")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   2359.3   2379.2  -1174.7   2349.3      387
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6539 -0.6449 -0.1219  0.5635  3.6676
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  1.949   1.396
##  year     (Intercept) 38.100   6.173
##  Residual             19.504   4.416
## Number of obs: 392, groups:  id, 64; year, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   59.706      2.330  25.623
## cmean_4       -2.058      1.884  -1.092
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 0.530
```

```r
summary(model1.3new)
```

```
## Cov prior  : id ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 3.8405
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 | id)
##    Data: subset(data_5yrs, period == "new")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   4652.0   4675.2  -2321.0   4642.0      765
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0099 -0.6201 -0.0536  0.5744  4.3799
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  4.396   2.097
##  year     (Intercept)  7.360   2.713
##  Residual             20.462   4.523
## Number of obs: 770, groups:  id, 99; year, 12
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  55.4824     0.8524  65.091
## cmean_4      -1.7136     0.5744  -2.983
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 -0.229
```

```
r.squaredGLMM(model1.3old)
```

```
##             R2m       R2c
## [1,] 0.07812982 0.6980824
```

```
r.squaredGLMM(model1.3new)
```

```
##            R2m       R2c
## [1,] 0.1450219 0.4569972
```

Compare with previous model using likelihood ratio test and AIC.

```
chi2 <- 2*(summary(model1.3old)$logLik - summary(model1.1old)$logLik)
1-pchisq(chi2, 1)
```

```
## 'log Lik.' 0.05333056 (df=5)
```

```
chi2 <- 2*(summary(model1.3new)$logLik - summary(model1.1new)$logLik)
1-pchisq(chi2, 1)
```

```
## 'log Lik.' 1.347611e-11 (df=5)
```

```r
AIC(model1.1old, model1.2old, model1.3old)
```

```
##             df      AIC
## model1.1old  4 2361.054
## model1.2old  5 2362.193
## model1.3old  5 2359.321
```

```r
AIC(model1.1new, model1.2new, model1.3new)
```

```
##             df      AIC
## model1.1new  4 4695.696
## model1.2new  5 4695.058
## model1.3new  5 4651.952
```

Linear fixed effects with linear random regression model

```r
model1.4old <- blmer(FFD ~ cmean_4 + (1|year) + (1+cmean_4|id), REML = FALSE,
                     data = subset(data_5yrs,period=="old"),
                     lmerControl(optimizer = "Nelder_Mead"))
model1.4new <- blmer(FFD ~ cmean_4 + (1|year) + (1+cmean_4|id), REML = FALSE,
                     data = subset(data_5yrs,period=="new"),
                     lmerControl(optimizer = "Nelder_Mead"))
summary(model1.4old)
```

```
## Cov prior  : id ~ wishart(df = 4.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 7.4859
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 + cmean_4 | id)
##    Data: subset(data_5yrs, period == "old")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   2363.7   2391.5  -1174.9   2349.7      385
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6123 -0.6459 -0.1077  0.5498  3.2025
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept)  2.4271  1.5579
##           cmean_4      0.6516  0.8072   0.50
##  year     (Intercept) 38.1240  6.1745
##  Residual             18.8274  4.3391
## Number of obs: 392, groups:  id, 64; year, 10
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   59.693      2.332  25.595
## cmean_4       -2.051      1.887  -1.087
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 0.531
```

```r
summary(model1.4new)
```

```
## Cov prior  : id ~ wishart(df = 4.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
##            : year ~ wishart(df = 3.5, scale = Inf, posterior.scale = cov, common.scale = TRUE)
## Prior dev  : 10.4712
##
## Linear mixed model fit by maximum likelihood  ['blmerMod']
## Formula: FFD ~ cmean_4 + (1 | year) + (1 + cmean_4 | id)
##    Data: subset(data_5yrs, period == "new")
## Control: lmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##   4634.6   4667.1  -2310.3   4620.6      763
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0701 -0.6142 -0.0652  0.6020  4.1189
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept)  3.7346  1.9325
##           cmean_4      0.7079  0.8414   0.82
##  year     (Intercept)  7.4467  2.7289
##  Residual             19.1165  4.3722
## Number of obs: 770, groups:  id, 99; year, 12
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  55.4458     0.8518  65.095
## cmean_4      -1.6794     0.5834  -2.879
##
## Correlation of Fixed Effects:
##         (Intr)
## cmean_4 -0.200
```

```r
r.squaredGLMM(model1.4old)
```

```
##             R2m       R2c
## [1,] 0.07759191 0.7088405
```

```r
r.squaredGLMM(model1.4new)
```

```
##            R2m      R2c
## [1,] 0.1391813 0.493125
```

Compare with previous model using likelihood ratio test and AIC.

```r
chi2 <- 2*(summary(model1.4old)$logLik - summary(model1.3old)$logLik)
# The df difference between models can be checked by
# looking at the df within the models being compared
summary(model1.3old)$logLik
```

```
## 'log Lik.' -1174.66 (df=5)
```

```r
summary(model1.4old)$logLik
```

```
## 'log Lik.' -1174.872 (df=7)
```

```r
# Note that between model1.3 and model1.4 there is a change of 2 df, so the
# pchisq change needs to be specified with 2 df rather than 1 as in previous comparisons.
1-pchisq(chi2, 2)
```

```
## 'log Lik.' 1 (df=7)
```

```r
chi2 <- 2*(summary(model1.4new)$logLik - summary(model1.3new)$logLik)
# The df difference between models can be checked by
# looking at the df within the models being compared
summary(model1.3new)$logLik
```

```
## 'log Lik.' -2320.976 (df=5)
```

```r
summary(model1.4new)$logLik
```

```
## 'log Lik.' -2310.291 (df=7)
```

```r
# Note that between model1.3 and model1.4 there is a change of 2 df, so the
# pchisq change needs to be specified with 2 df rather than 1 as in previous comparisons.
1-pchisq(chi2, 2)
```

```
## 'log Lik.' 2.287776e-05 (df=7)
```

```r
AIC(model1.1old, model1.2old, model1.3old, model1.4old)
```

```
##             df      AIC
## model1.1old  4 2361.054
## model1.2old  5 2362.193
## model1.3old  5 2359.321
## model1.4old  7 2363.745
```

```r
AIC(model1.1new, model1.2new, model1.3new, model1.4new)
```

```
##             df      AIC
## model1.1new  4 4695.696
## model1.2new  5 4695.058
## model1.3new  5 4651.952
## model1.4new  7 4634.581
```

For the new period, model1.4 seems to be the best model, but not for the old period (where model1.3 is the best one).

Extract BLUPs from model1.4 (linear random regression mixed model)

BLUPs represent the response of a given id to the fixed effect of temperature as the difference between that id's predicted response and the population-level average predicted response. Here, we calculate and plot BLUPs for ranking plasticity.

```
id_blups_old <- ranef(model1.4old)$`id`
id_blups_new <- ranef(model1.4new)$`id`
id_index_old <- as.factor(c(1:64))
id_index_new <- as.factor(c(1:99))
id_data_old  <- cbind(id_index_old, id_blups_old)
id_data_new  <- cbind(id_index_new, id_blups_new)
colnames(id_data_old) <- c("id", "BLUP_int", "BLUP_slope")
colnames(id_data_new) <- c("id", "BLUP_int", "BLUP_slope")
id_data_oldnew<-rbind(id_data_old,id_data_new)
id_data_oldnew<-id_data_oldnew%>%
  rownames_to_column()%>%
  select(rowname,BLUP_int,BLUP_slope)%>%
  rename(id = rowname)%>%
  mutate(period=ifelse(grepl("old",id),"old","new"))
with(id_data_oldnew,cor(BLUP_int,BLUP_slope)) # highly correlated!
```

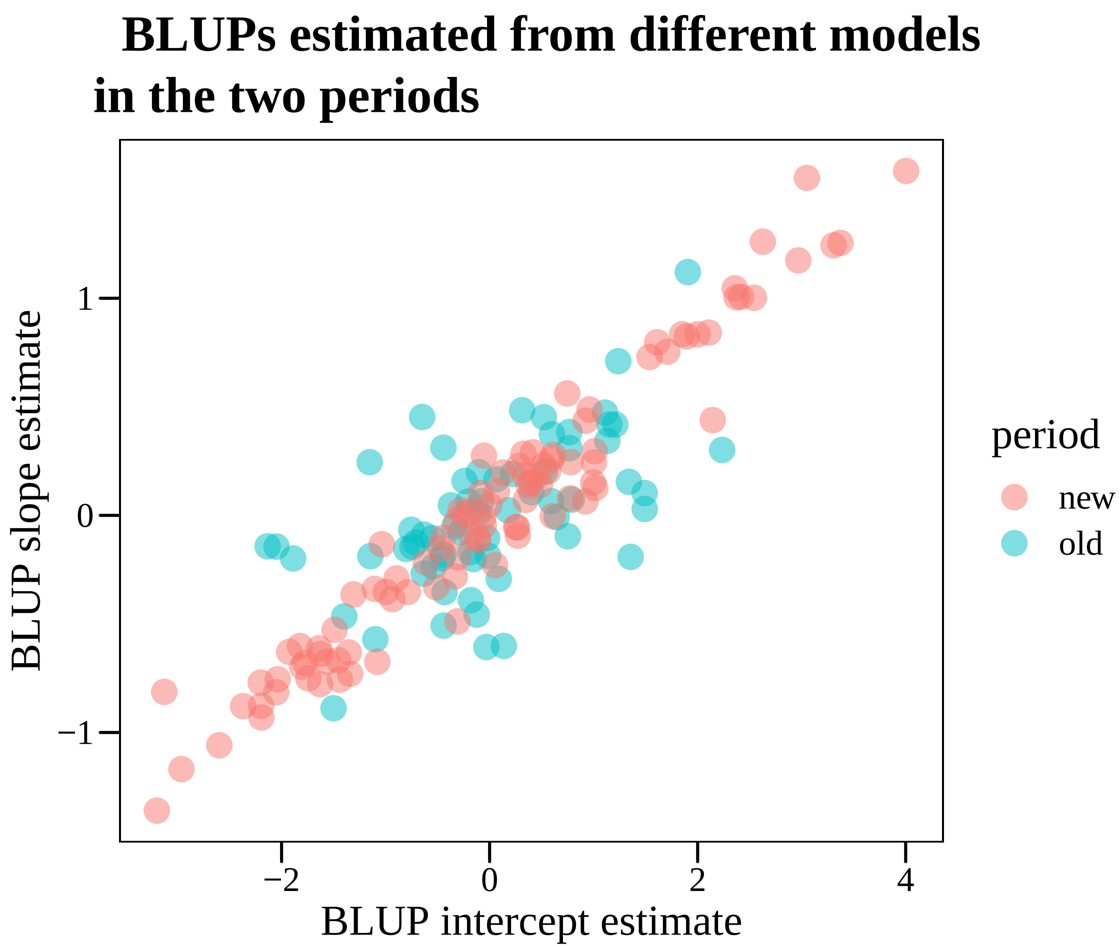
```
## [1] 0.904976
```

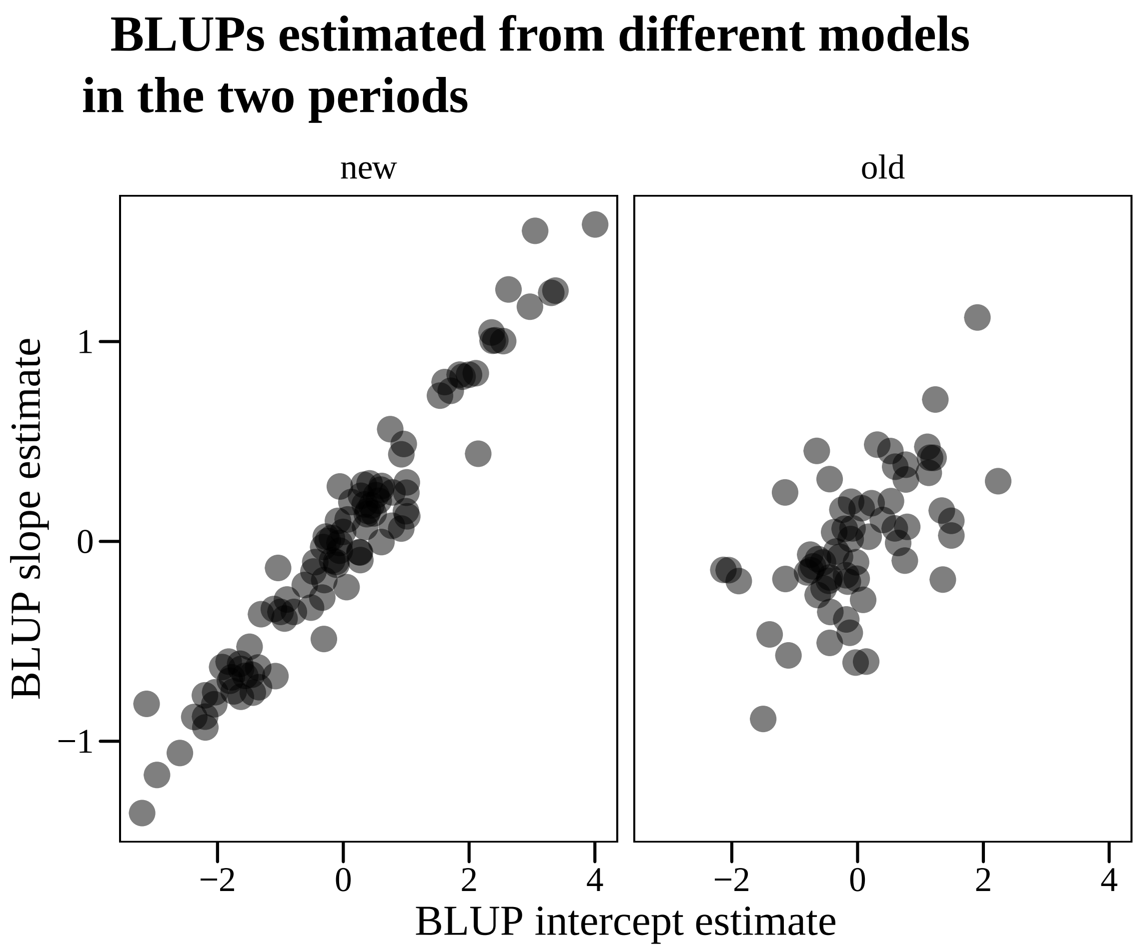
```
with(subset(id_data_oldnew,period=="old"),cor(BLUP_int,BLUP_slope)) # correlated!
```

```
## [1] 0.5779531
```

```
with(subset(id_data_oldnew,period=="new"),cor(BLUP_int,BLUP_slope)) # highly correlated!
```

```
## [1] 0.9735763
```


BLUPs estimated from different models
in the two periods
BLUP slope estimate
BLUP intercept estimate
1
0
−1
−2
0
2
4
period
new
old



MCMCglmm models (global)

Code based on Arnold et al. 2019 Phil. Trans. R. Soc. B.

Fitting bivariate models of fitness and FFD, with random regressions for individuals, using a Poisson distribution for fitness, instead of Gaussian (and absolute instead of relative fitness). Using mean April temperature and individuals with at least 5 years of data. Using either mean fitness per year of life or mean fitness per flowering event. Including / not including mean shoot volume over all years with available data (with an effect on fitness) as a condition variable.

Data preparation

```
data_5yrs_total<-data_5yrs %>%
  group_by(id)%>%
  summarise(mean_fitness_life=sum(n_intact_seeds)/mean(n_years_life),
            # Mean fitness per living year = sum fitness / n years alive
            mean_fitness_fl=sum(n_intact_seeds)/mean(n_years_fl))%>%
  # Mean fitness per fl event = sum fitness / n years when each id flowered
   arrange(.,id) # Order by id

 # Get values of shoot volume for all ids/years (incl also yrs when not fl)
shoot_vol<-alldata[c(1,3,14)]%>%
```

```r
  arrange(id)%>%                    # Order by id
  filter(!is.na(shoot_vol))%>%     # Remove NAs
  filter(id %in% data_5yrs$id)%>%
  droplevels()

shoot_vol_means<-shoot_vol%>%
  group_by(id)%>%
  summarise(shoot_vol_mean=mean(shoot_vol)) # Mean of all available values

# Join shoot volume data
(data_5yrs_total<-data_5yrs_total%>%right_join(shoot_vol_means))
```

```
## # A tibble: 163 x 4
##    id      mean_fitness_life mean_fitness_fl shoot_vol_mean
##    <fct>               <dbl>           <dbl>          <dbl>
##  1 new_10             14.3            15.7            9794.
##  2 new_100             3.89            5.83           1959.
##  3 new_101             2.25            3.00           1195.
##  4 new_102             5.61            6.73           3269.
##  5 new_103             3.60            4.32           1694.
##  6 new_104             1.58            2.71           1056.
##  7 new_106             1.74            2.98           1972.
##  8 new_107             3               4              1108.
##  9 new_108             1.17            2.01            755.
## 10 new_109             0.165           0.180          2406.
## # ... with 153 more rows
```

```r
 with(data_5yrs_total,cor(mean_fitness_life,mean_fitness_fl))  # Highly correlated
```

```
## [1] 0.9557609
```

Mean fitness per year of life

With no condition variable

Stack data:

```r
# Create a single data-set "data.stack10", with single column at start
# to index observations
data.stack10 <- c()
data.stack10$Obs <- 1:(163 + 1162)
data.stack10$id <- c(as.character(data_5yrs_total$id),as.character(data_5yrs$id))
# ids in alphabetical order

# Add first_yr to total data +
# Year column is only relevant for FFD, but is set to first_yr for fitness values
data_5yrs_total$first_yr<-ifelse(grepl("old",as.character(data_5yrs_total$id)),1987,2006)

data.stack10$year <- c(data_5yrs_total$first_yr,
                     data_5yrs$year)
```

```r
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack10$temp <- c(rep(0, 163), data_5yrs$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack10$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness_life), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack10$traits <- c(rep("fitness", 163), rep("FFD", 1162))
data.stack10$variable <- data.stack10$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack10$family <- c(rep("poisson", 163), rep("gaussian", 1162))
data.stack10 <- data.frame(data.stack10)

data.stack10$id <- as.factor(data.stack10$id)
data.stack10$year <- as.factor(data.stack10$year)
head(data.stack10)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                14 fitness  fitness poisson
## 2   2 new_100 2006    0                 4 fitness  fitness poisson
## 3   3 new_101 2006    0                 2 fitness  fitness poisson
## 4   4 new_102 2006    0                 6 fitness  fitness poisson
## 5   5 new_103 2006    0                 4 fitness  fitness poisson
## 6   6 new_104 2006    0                 2 fitness  fitness poisson
```

```r
# Scaling factor for MCMCglmm iterations
sc <- 1000 # Increase this parameter for longer runs

priorBiv_RR10 <- list(G = list(G1 = list(V = diag(1), nu = 1)),
                    # ^ random effect for year (fitted for FFD only)
                    R = list(R1 = list(V = diag(3), nu = 3, covu = TRUE),
                             # ^ 3-way var-cov matrix of (id + temp:id) for FFD,
                             # residual for fitness
                             R2 = list(V = diag(1), nu = 1))) # residual for FFD

modelBV_RR10 <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivdiuals in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivdiuals between years
                       # (labelled by 'Obs')
                       data = data.stack10,
                       prior = priorBiv_RR10,
```

```
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(modelBV_RR10,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_R
```

```
kable(summary(modelBV_RR10)$solutions,digits=c(3,3,3,0,3),caption="Fixed effects")
```

Table 1: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	57.424	55.405	59.487	2000	0.000
variablefitness	1.161	1.025	1.309	2000	0.000
at.level(variable, “FFD”):temp	-2.405	-4.057	-0.904	2000	0.004

```
kable(summary(modelBV_RR10)$Gcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 2: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”):at.level(variable, “FFD”).year	25.26	11.466	42.674	2000

```
kable(summary(modelBV_RR10)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 3: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	3.116	1.887	4.481	1799
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	1.012	0.501	1.674	2000
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.554	-0.906	-0.216	2000
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	1.012	0.501	1.674	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	0.795	0.372	1.247	2148
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	-0.206	-0.411	-0.013	1636
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.554	-0.906	-0.216	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	-0.206	-0.411	-0.013	1636
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	0.485	0.316	0.652	2000
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	19.290	17.543	20.980	2000

```
kable(diag(autocorr(modelBV_RR10$Sol)[2, , ]),caption="Autocorrelation")
```

Table 4: Autocorrelation

	x
variableFFD	0.0147066
variablefitness	0.0102773

	x
at.level(variable, “FFD”):temp	0.0140045

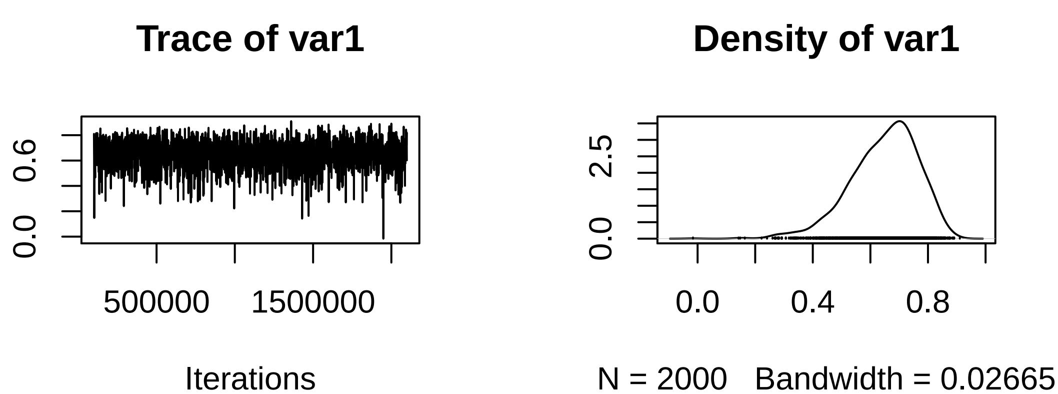
```r
kable(diag(autocorr(modelBV_RR10$VCV)[2, , ]),caption="Autocorrelation")
```

Table 5: Autocorrelation

	x
at.level(variable, “FFD”):at.level(variable, “FFD”).year	0.0160400
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	0.0288033
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	-0.0188162
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	0.0103356
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	-0.0188162
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	-0.0359228
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	0.0014792
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	0.0103356
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	0.0014792
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	-0.0255814
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	0.0081337

Among-individual correlation between intercepts and slopes for FFD:

```r
cor_BV_RR_10_intslope <-
  modelBV_RR10$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"]/
(sqrt(modelBV_RR10$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"])*
sqrt(modelBV_RR10$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_10_intslope)
```




```r
posterior.mode(cor_BV_RR_10_intslope)
```

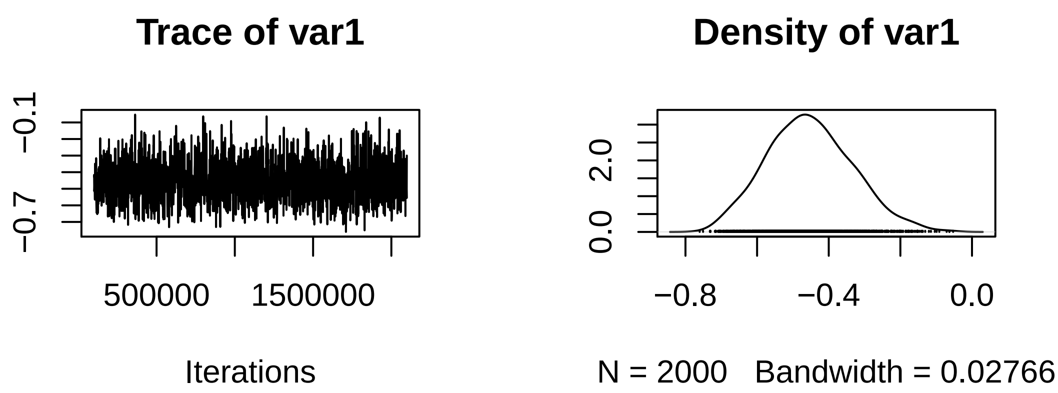
```
##      var1
## 0.7020806
```

```
HPDinterval(cor_BV_RR_10_intslope)
```

```
##          lower     upper
## var1 0.4223704 0.8552794
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
cor_BV_RR_10_intfit <-
  modelBV_RR10$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\").id"]/
  (sqrt(modelBV_RR10$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR10$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"]))
plot(cor_BV_RR_10_intfit)
```




```
posterior.mode(cor_BV_RR_10_intfit)
```

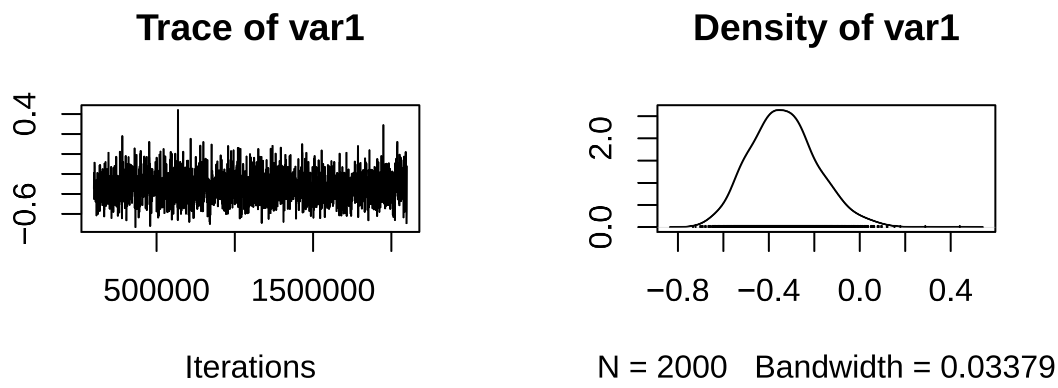
```
##       var1
## -0.4707792
```

```
HPDinterval(cor_BV_RR_10_intfit)
```

```
##           lower      upper
## var1 -0.6948572 -0.2335608
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between fitness and variation in slopes for FFD:

```
cor_BV_RR_10_slopefit <-
  modelBV_RR10$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]/
  (sqrt(modelBV_RR10$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR10$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_10_slopefit)
```




```r
posterior.mode(cor_BV_RR_10_slopefit)
```

```
##       var1
## -0.2621944
```

```r
HPDinterval(cor_BV_RR_10_slopefit)
```

```
##           lower       upper
## var1 -0.6387469 -0.06160169
## attr(,"Probability")
## [1] 0.95
```

Intercepts and slopes of RNs are positively correlated: Plants that flower earlier on average are also more responsive to temperature. Fitness is negatively correlated with both the intercept and the slope of the RN: When temperature increases, individuals that flower earlier on average and are more responsive to temperature have higher fitness (is this really what the results indicate?).

Extract selection coefficients Selection differentials or gradients should be calculated using relative fitness, and models are typically fitted assuming Gaussian errors. However, where the fitness measure follows a non-Gaussian distribution, as is typically the case with skewed distributions of fitness, a GLMM of absolute fitness will be preferable. The resulting covariances returned by the model will then be between the trait on the data scale and fitness on a 'latent' (link-function) scale. These estimates need to be transformed if data-scale estimates of selection are required. However, in the case of a GLMM with a log-link function (e.g. Poisson, over-dispersed Poisson, or negative binomial distribution), it is possible to exploit the fact that the latent-scale covariance with absolute fitness is equivalent to the data-scale covariance of relative fitness: consequently, and conveniently, the covariance components of Pind on the latent scale can simply be treated as selection differentials S. By extension, estimates of b as indicated above will also provide data-scale selection gradients.

```r
# Extract 3x3 matrix of variance-covariance values for intercepts and slopes
# of temp, and fitness
# These are in the 2nd-10th columns of model output
P.modelBV_RR10 <- modelBV_RR10$VCV[,2:10]
P.modelBV_RR10.mode <- matrix(1:9, nrow = 3)
for (k in 1:9) P.modelBV_RR10.mode[k] <- posterior.mode(P.modelBV_RR10[,k])
P.modelBV_RR10.mode
```

```
##             [,1]       [,2]       [,3]
## [1,]  2.7650555  0.8370081 -0.5515958
## [2,]  0.8370081  0.6945021 -0.1948936
## [3,] -0.5515958 -0.1948936  0.4656639
```

```
# Extract selection *differentials* (i.e. covariances) for intercept and slope:
# and calculate posterior mode and credible intervals for each
S.modelBV_RR10 <- modelBV_RR10$VCV[, c(4,7)]
S.modelBV_RR10 <- P.modelBV_RR10[, c(3,6)] # This is exactly the same as above
colnames(S.modelBV_RR10) <- c("S_intercepts", "S_slopes")
S.modelBV_RR10.mode <- P.modelBV_RR10.mode[1:2, 3]
S.modelBV_RR10.mode
```

```
## [1] -0.5515958 -0.1948936
```

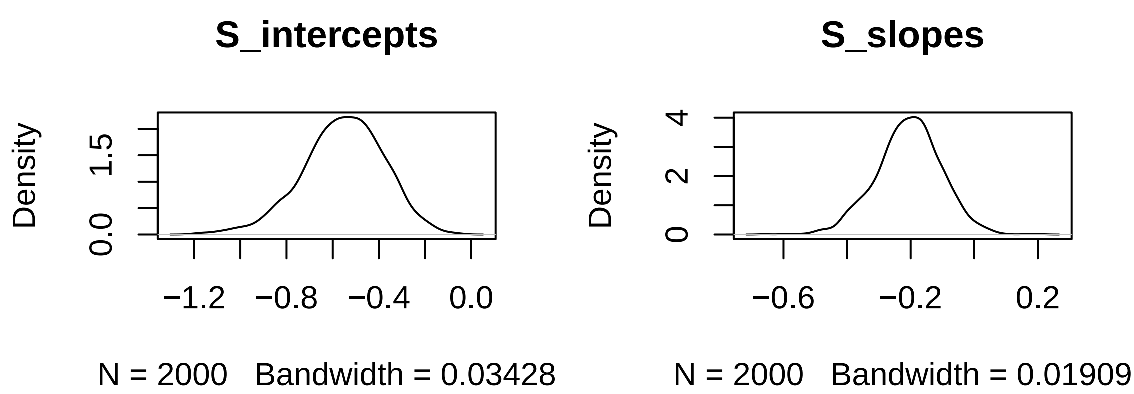
```
posterior.mode(mcmc(S.modelBV_RR10)) # This is exactly the same as above
```

```
## S_intercepts     S_slopes
##   -0.5515958   -0.1948936
```

```
HPDinterval(mcmc(S.modelBV_RR10))
```

```
##                   lower       upper
## S_intercepts -0.9057789 -0.21572385
## S_slopes     -0.4110405 -0.01343359
## attr(,"Probability")
## [1] 0.95
```

```
# Plot posterior distribution of selection differentials
par(mfrow = c(1,2))
plot(density(S.modelBV_RR10[,1]), main = "S_intercepts")
plot(density(S.modelBV_RR10[,2]), main = "S_slopes")
```



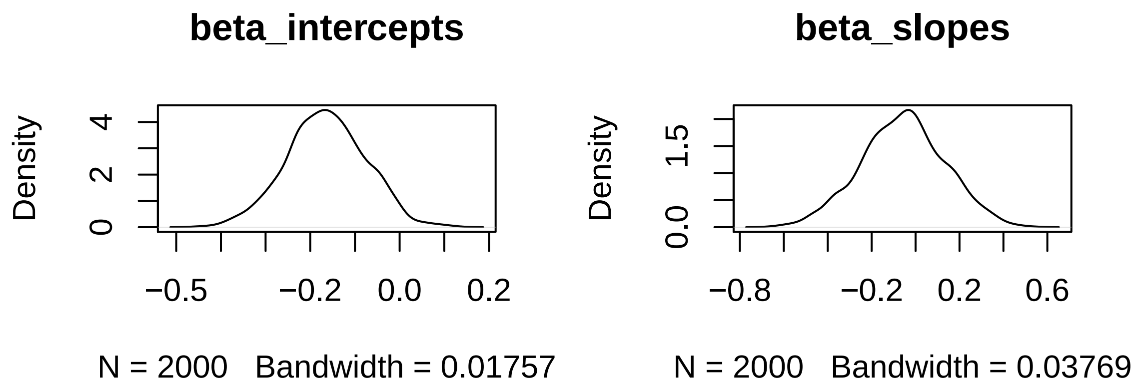
```r
# Estimate selection gradients for intercept and slope (beta = S / P)
# on each sample of posterior and extract their mode
n <- length(modelBV_RR10$VCV[,2])   # sample size
beta_post_RR10 <- matrix(NA, n ,2)

for (i in 1:n) {
  P3 <- matrix(rep(NA, 9), nrow = 3)
  # 3x3 matrix of var-cov for individual X.int, X.slope and fitness
  for (k in 1:9) {P3[k] <- P.modelBV_RR10[i, k] }
  P2 <- P3[1:2, 1:2]   # 2x2 matrix of just trait intercept & slope var-cov
  S <- P3[1:2, 3]   # selection differentials on traits (last column of P3)
  beta_post_RR10[i,] <- solve(P2) %*% S   # selection gradients beta = P^-1 * S
}

# Finally, extract and plot the selection gradients posterior modes
# and 95% credible intervals for both selection on intercepts (trait value)
# and slopes (trait plasticity).
# Note that credible intervals are not exactly confidence intervals. See here:
# https://statsdirect.com/help/basics/confidence_interval.htm and
# https://stats.stackexchange.com/questions/2272/

colnames(beta_post_RR10) <- c("beta_intercepts", "beta_slopes")
posterior.mode(mcmc(beta_post_RR10))
```

```
## beta_intercepts     beta_slopes
##     -0.16607681     -0.02959425
```

```r
HPDinterval(mcmc(beta_post_RR10))
```

```
##                      lower        upper
## beta_intercepts -0.3427632 -0.000675528
## beta_slopes     -0.4055639  0.350744907
## attr(,"Probability")
## [1] 0.95
```

```r
# Plot posterior distribution of selection gradients
par(mfrow = c(1,2))
plot(density(beta_post_RR10[,1]), main = "beta_intercepts")
plot(density(beta_post_RR10[,2]), main = "beta_slopes")
```




```
# NB selection differentials and gradients here are from covariances
# with latent-scale absolute fitness
# These are equivalent to covariances with data-scale relative fitness:
# see main text of paper
```

The selection differentials are “significant” for both RN intercepts and slopes, but the selection gradients are not “significant” for any of them. This means that, there is significant total selection (direct + indirect) on intercepts and slopes, but after correcting for the covariance between them, there is no direct selection on any of them.

With shoot volume

Stack data:

```
# Create a single data-set "data.stack14", with single column at start
# to index observations
data.stack14 <- c()
data.stack14$Obs <- 1:(163 + 1162)
data.stack14$id <- c(as.character(data_5yrs_total$id),as.character(data_5yrs$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack14$year <- c(data_5yrs_total$first_yr,
                       data_5yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack14$temp <- c(rep(0, 163), data_5yrs$cmean_4)

# Shoot volume column is only relevant for fitness, but is set to 0 for FFD values
# Using log of mean shoot volume over all years when available, centered
data_5yrs_total<-data_5yrs_total%>%
  mutate(shoot_vol_mean_log=log(shoot_vol_mean),
         cn_shoot_vol_mean_log=scale(shoot_vol_mean_log,center=T,scale=F))
data.stack14$cn_shoot_vol <- c(data_5yrs_total$cn_shoot_vol_mean_log,rep(0, 1162))

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack14$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness_life), data_5yrs$FFD)
```

```r
# Create 3 index columns needed for MCMCglmm
data.stack14$traits <- c(rep("fitness", 163), rep("FFD", 1162))
data.stack14$variable <- data.stack14$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack14$family <- c(rep("poisson", 163), rep("gaussian", 1162))
data.stack14 <- data.frame(data.stack14)

data.stack14$id <- as.factor(data.stack14$id)
data.stack14$year <- as.factor(data.stack14$year)
head(data.stack14)
```

```
##   Obs      id year temp cn_shoot_vol fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0    1.9543895                14 fitness  fitness poisson
## 2   2 new_100 2006    0    0.3448666                 4 fitness  fitness poisson
## 3   3 new_101 2006    0   -0.1489826                 2 fitness  fitness poisson
## 4   4 new_102 2006    0    0.8572416                 6 fitness  fitness poisson
## 5   5 new_103 2006    0    0.1998053                 4 fitness  fitness poisson
## 6   6 new_104 2006    0   -0.2730598                 2 fitness  fitness poisson
```

```r
modelBV_RR14 <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp +  # single fixed effect of temp
                           at.level(variable,"fitness"):cn_shoot_vol,
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduials in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack14,
                       prior = priorBiv_RR10,
                       family = NULL, # specified already in the data-set
                       nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                       verbose = F,singular.ok = T)
save(modelBV_RR14,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_RR
```

```r
kable(summary(modelBV_RR14)$solutions,digits=c(3,3,3,0,3),caption="Fixed effects")
```

Table 6: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	57.404	55.268	59.666	2000	0.000
variablefitness	1.156	1.008	1.272	1833	0.000
at.level(variable, “FFD”):temp	-2.382	-3.980	-0.700	1864	0.009
at.level(variable, “fitness”):cn_shoot_vol	0.521	0.308	0.725	2000	0.000

```r
kable(summary(modelBV_RR14)$Gcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 7: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”):at.level(variable, “FFD”).year	25.447	11.77	41.81	2000

```r
kable(summary(modelBV_RR14)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 8: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	3.157	1.831	4.494	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	1.035	0.442	1.623	2000
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.283	-0.630	0.055	2000
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	1.035	0.442	1.623	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	0.804	0.375	1.245	2000
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	-0.079	-0.271	0.104	2000
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.283	-0.630	0.055	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	-0.079	-0.271	0.104	2000
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	0.368	0.240	0.503	2000
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	19.253	17.504	20.962	2000

```r
kable(diag(autocorr(modelBV_RR14$Sol)[2, , ]),caption="Autocorrelation")
```

Table 9: Autocorrelation

	x
variableFFD	0.0080180
variablefitness	-0.0077778
at.level(variable, “FFD”):temp	0.0349576
at.level(variable, “fitness”):cn_shoot_vol	-0.0191315

```r
kable(diag(autocorr(modelBV_RR14$VCV)[2, , ]),caption="Autocorrelation")
```

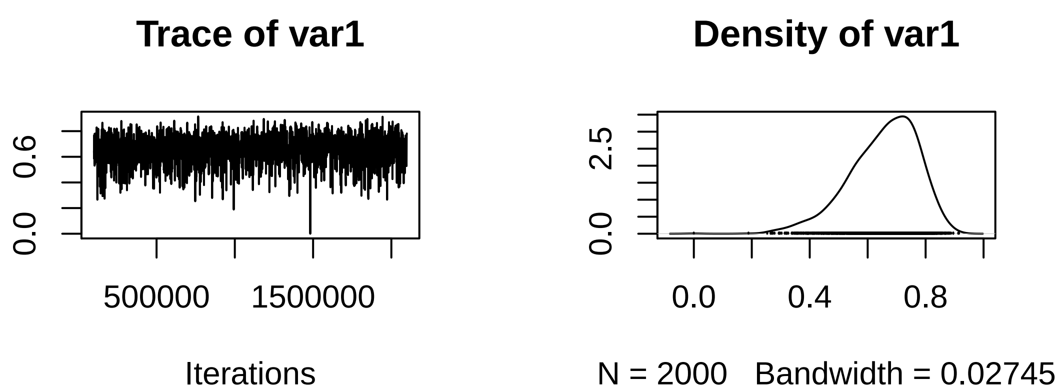
Table 10: Autocorrelation

	x
at.level(variable, “FFD”):at.level(variable, “FFD”).year	0.0053224
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	-0.0234186
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	-0.0261320
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	0.0064403
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	-0.0261320
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	-0.0042325
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	0.0107429
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	0.0064403
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	0.0107429

	x
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	-0.0049482
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	0.0136563

Among-individual correlation between intercepts and slopes for FFD:

```
cor_BV_RR_14_intslope <-
  modelBV_RR14$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"]/
(sqrt(modelBV_RR14$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"])*
sqrt(modelBV_RR14$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_14_intslope)
```




```
posterior.mode(cor_BV_RR_14_intslope)
```

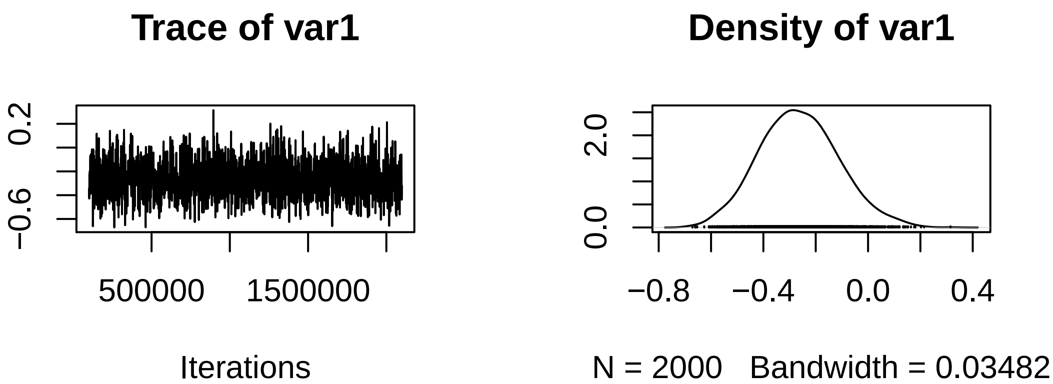
```
##      var1
## 0.7384523
```

```
HPDinterval(cor_BV_RR_14_intslope)
```

```
##          lower     upper
## var1 0.4090851 0.8568558
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
cor_BV_RR_14_intfit <-
  modelBV_RR14$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\").id"]/
  (sqrt(modelBV_RR14$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR14$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"]))
plot(cor_BV_RR_14_intfit)
```




```r
posterior.mode(cor_BV_RR_14_intfit)
```

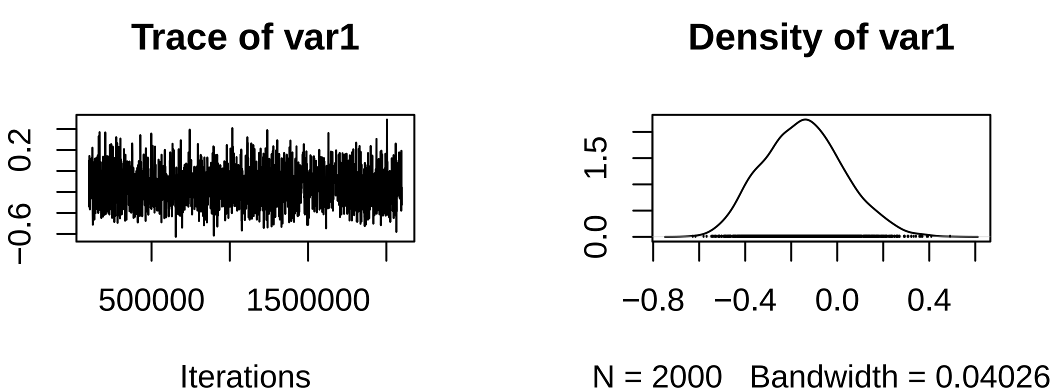
```
##      var1
## -0.310966
```

```r
HPDinterval(cor_BV_RR_14_intfit)
```

```
##          lower      upper
## var1 -0.575524 0.01667554
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between fitness and variation in slopes for FFD:

```r
cor_BV_RR_14_slopefit <-
  modelBV_RR14$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]/
  (sqrt(modelBV_RR14$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR14$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_14_slopefit)
```



```
posterior.mode(cor_BV_RR_14_slopefit)
```

```
##       var1
## -0.2371037
```

```
HPDinterval(cor_BV_RR_14_slopefit)
```

```
##           lower    upper
## var1 -0.4865946 0.178697
## attr(,"Probability")
## [1] 0.95
```

Intercepts and slopes of RNs are positively correlated: Plants that flower earlier on average are also more responsive to temperature. Fitness is not correlated with either the intercept or the slope of the RN: There is no selection on either the intercept or the slope of the RN when including plant size as a condition variable.

```
P.modelBV_RR14 <- modelBV_RR14$VCV[,2:10]
P.modelBV_RR14.mode <- matrix(1:9, nrow = 3)
for (k in 1:9) P.modelBV_RR14.mode[k] <- posterior.mode(P.modelBV_RR14[,k])
P.modelBV_RR14.mode
```

Extract selection coefficients

```
##            [,1]        [,2]        [,3]
## [1,]  3.1845808  1.07871115 -0.25076291
## [2,]  1.0787111  0.70900207 -0.07611192
## [3,] -0.2507629 -0.07611192  0.33449678
```

```
S.modelBV_RR14 <- modelBV_RR14$VCV[, c(4,7)]
S.modelBV_RR14 <- P.modelBV_RR14[, c(3,6)]
colnames(S.modelBV_RR14) <- c("S_intercepts", "S_slopes")
S.modelBV_RR14.mode <- P.modelBV_RR14.mode[1:2, 3]
S.modelBV_RR14.mode
```

```
## [1] -0.25076291 -0.07611192
```

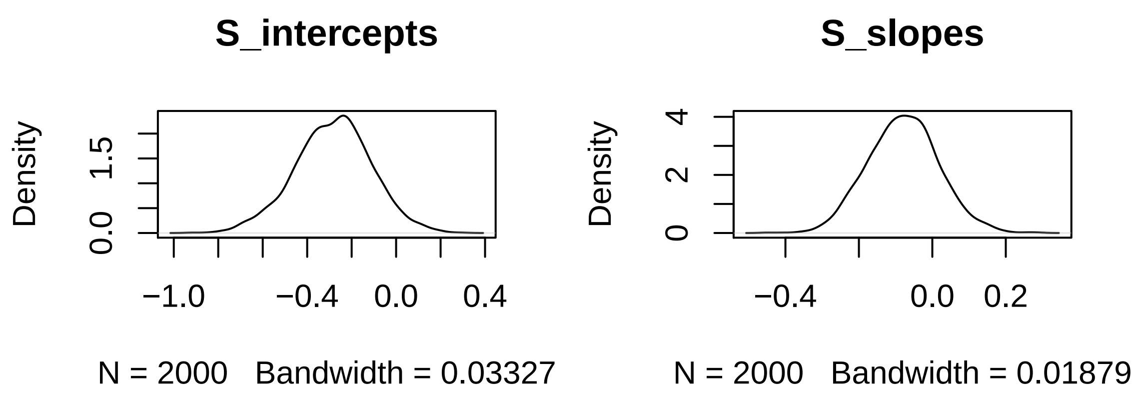
```
posterior.mode(mcmc(S.modelBV_RR14))
```

```
## S_intercepts     S_slopes
##  -0.25076291  -0.07611192
```

```
HPDinterval(mcmc(S.modelBV_RR14))
```

```
##                   lower      upper
## S_intercepts -0.6296267 0.05480024
## S_slopes     -0.2710516 0.10384443
## attr(,"Probability")
## [1] 0.95
```

```r
par(mfrow = c(1,2))
plot(density(S.modelBV_RR14[,1]), main = "S_intercepts")
plot(density(S.modelBV_RR14[,2]), main = "S_slopes")
```



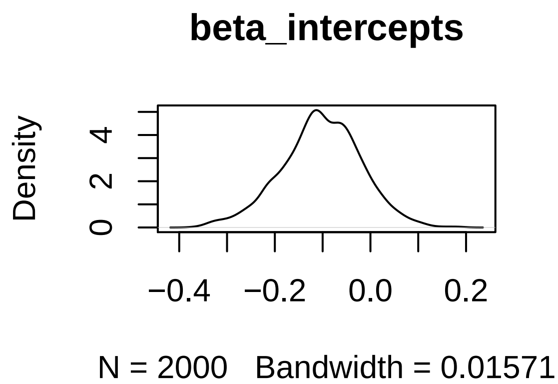

```r
n <- length(modelBV_RR14$VCV[,2])   # sample size
beta_post_RR14 <- matrix(NA, n ,2)

for (i in 1:n) {
  P3 <- matrix(rep(NA, 9), nrow = 3)
  # 3x3 matrix of var-cov for individual X.int, X.slope and LBS
  for (k in 1:9) {P3[k] <- P.modelBV_RR14[i, k] }
  P2 <- P3[1:2, 1:2]    # 2x2 matrix of just trait intercept & slope var-cov
  S <- P3[1:2, 3]    # selection differentials on traits (last column of P3)
  beta_post_RR14[i,] <- solve(P2) %*% S   # selection gradients beta = P^-1 * S
}

colnames(beta_post_RR14) <- c("beta_intercepts", "beta_slopes")
posterior.mode(mcmc(beta_post_RR14))
```

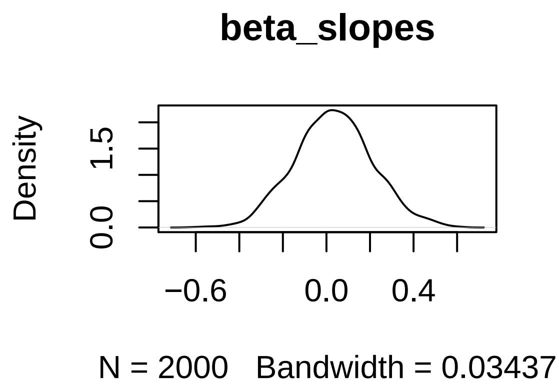
```
## beta_intercepts     beta_slopes
##     -0.11118721      0.01542094
```

```r
HPDinterval(mcmc(beta_post_RR14))
```

```
##                      lower      upper
## beta_intercepts -0.2666901 0.06590026
## beta_slopes     -0.3256418 0.36715303
## attr(,"Probability")
## [1] 0.95
```

```r
par(mfrow = c(1,2))
plot(density(beta_post_RR14[,1]), main = "beta_intercepts")
plot(density(beta_post_RR14[,2]), main = "beta_slopes")
```





The selection differentials and gradients are not "significant" for either RN intercepts or slopes. This means that there no significant selection (either direct or indirect) on intercepts and slopes of the RNs.

Mean fitness per flowering event

With no condition variable

Stack data:

```
# Create a single data-set "data.stack12", with single column at start
# to index observations
data.stack12 <- c()
data.stack12$Obs <- 1:(163 + 1162)
data.stack12$id <- c(as.character(data_5yrs_total$id),as.character(data_5yrs$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack12$year <- c(data_5yrs_total$first_yr,
                       data_5yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack12$temp <- c(rep(0, 163), data_5yrs$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack12$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness_fl), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack12$traits <- c(rep("fitness", 163), rep("FFD", 1162))
data.stack12$variable <- data.stack12$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack12$family <- c(rep("poisson", 163), rep("gaussian", 1162))
data.stack12 <- data.frame(data.stack12)

data.stack12$id <- as.factor(data.stack12$id)
data.stack12$year <- as.factor(data.stack12$year)
head(data.stack12)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                16 fitness  fitness poisson
## 2   2 new_100 2006    0                 6 fitness  fitness poisson
## 3   3 new_101 2006    0                 3 fitness  fitness poisson
## 4   4 new_102 2006    0                 7 fitness  fitness poisson
## 5   5 new_103 2006    0                 4 fitness  fitness poisson
## 6   6 new_104 2006    0                 3 fitness  fitness poisson
```

```
modelBV_RR12 <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack12,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
save(modelBV_RR12,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_R
```

```
kable(summary(modelBV_RR12)$solutions,digits=c(3,3,3,0,3),caption="Fixed effects")
```

Table 11: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	57.368	55.154	59.478	2000	0.000
variablefitness	1.512	1.372	1.640	2000	0.000
at.level(variable, "FFD"):temp	-2.436	-3.946	-0.754	2000	0.007

```
kable(summary(modelBV_RR12)$Gcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 12: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, "FFD"):at.level(variable, "FFD").year	25.829	11.249	43.351	1594

```
kable(summary(modelBV_RR12)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 13: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, "FFD").id:at.level(variable, "FFD").id	3.136	1.841	4.430	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	1.027	0.505	1.640	2000
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.505	-0.863	-0.181	2000
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	1.027	0.505	1.640	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	0.801	0.406	1.280	2000
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.179	-0.386	0.007	2000
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.505	-0.863	-0.181	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.179	-0.386	0.007	2000
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.487	0.346	0.650	2000
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	19.272	17.666	21.023	2000

```
kable(diag(autocorr(modelBV_RR12$Sol)[2, , ]),caption="Autocorrelation")
```

Table 14: Autocorrelation

	x
variableFFD	-0.0108249
variablefitness	0.0069810
at.level(variable, "FFD"):temp	-0.0259325

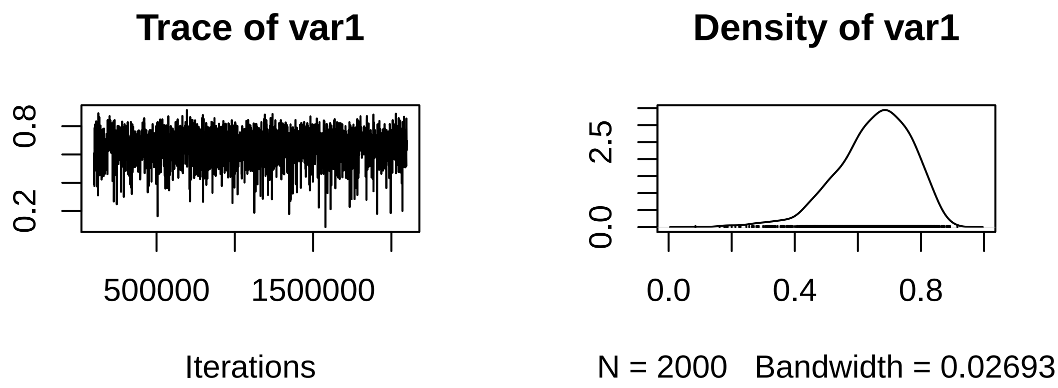
```
kable(diag(autocorr(modelBV_RR12$VCV)[2, , ]),caption="Autocorrelation")
```

Table 15: Autocorrelation

	x
at.level(variable, "FFD"):at.level(variable, "FFD").year	0.0301373
at.level(variable, "FFD").id:at.level(variable, "FFD").id	-0.0115915
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	-0.0134405
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.0136755
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	-0.0134405
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	0.0136845
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	0.0129590
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.0136755
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	0.0129590
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.0042579
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	-0.0157954

Among-individual correlation between intercepts and slopes for FFD:

```
cor_BV_RR_12_intslope <-
  modelBV_RR12$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"]/
(sqrt(modelBV_RR12$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"])*
sqrt(modelBV_RR12$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_12_intslope)
```




```
posterior.mode(cor_BV_RR_12_intslope)
```

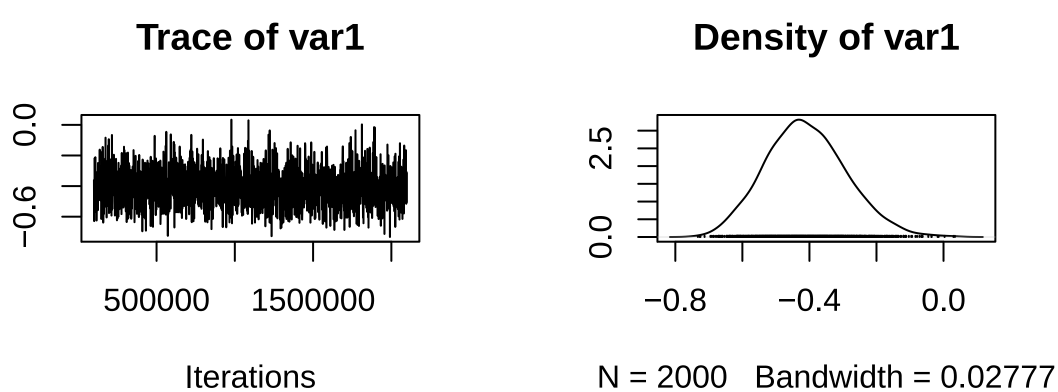
```
##      var1
## 0.6943047
```

```
HPDinterval(cor_BV_RR_12_intslope)
```

```
##          lower     upper
## var1 0.4238748 0.8514316
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
cor_BV_RR_12_intfit <-
  modelBV_RR12$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\").id"]/
  (sqrt(modelBV_RR12$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR12$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"]))
plot(cor_BV_RR_12_intfit)
```



```r
posterior.mode(cor_BV_RR_12_intfit)
```

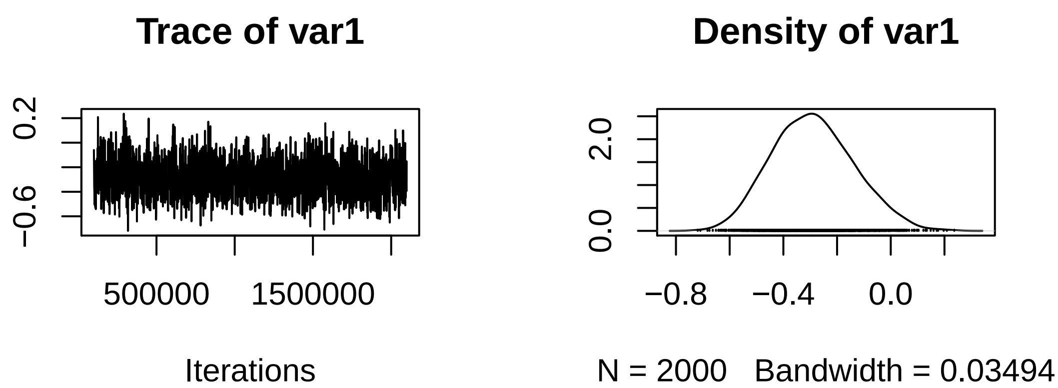
```
##       var1
## -0.4234863
```

```r
HPDinterval(cor_BV_RR_12_intfit)
```

```
##           lower      upper
## var1 -0.6479463 -0.1832756
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between fitness and variation in slopes for FFD:

```r
cor_BV_RR_12_slopefit <-
  modelBV_RR12$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]/
  (sqrt(modelBV_RR12$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR12$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_12_slopefit)
```




```r
posterior.mode(cor_BV_RR_12_slopefit)
```

```
##       var1
## -0.2954267
```

```r
HPDinterval(cor_BV_RR_12_slopefit)
```

```
##           lower        upper
## var1 -0.5834101 -0.002645371
## attr(,"Probability")
## [1] 0.95
```

Intercepts and slopes of RNs are positively correlated: Plants that flower earlier on average are also more responsive to temperature. Fitness is negatively correlated with both the intercept and the slope of the RN: When temperature increases, individuals that flower earlier on average and are more responsive to temperature have higher fitness (is this really what the results indicate?).

```r
P.modelBV_RR12 <- modelBV_RR12$VCV[,2:10]
P.modelBV_RR12.mode <- matrix(1:9, nrow = 3)
for (k in 1:9) P.modelBV_RR12.mode[k] <- posterior.mode(P.modelBV_RR12[,k])
P.modelBV_RR12.mode
```

Extract selection coefficients

```
##            [,1]       [,2]       [,3]
## [1,]  2.9378683  0.9398687 -0.4346212
## [2,]  0.9398687  0.6728728 -0.1827471
## [3,] -0.4346212 -0.1827471  0.4626394
```

```r
S.modelBV_RR12 <- modelBV_RR12$VCV[, c(4,7)]
S.modelBV_RR12 <- P.modelBV_RR12[, c(3,6)]
colnames(S.modelBV_RR12) <- c("S_intercepts", "S_slopes")
S.modelBV_RR12.mode <- P.modelBV_RR12.mode[1:2, 3]
S.modelBV_RR12.mode
```

```
## [1] -0.4346212 -0.1827471
```

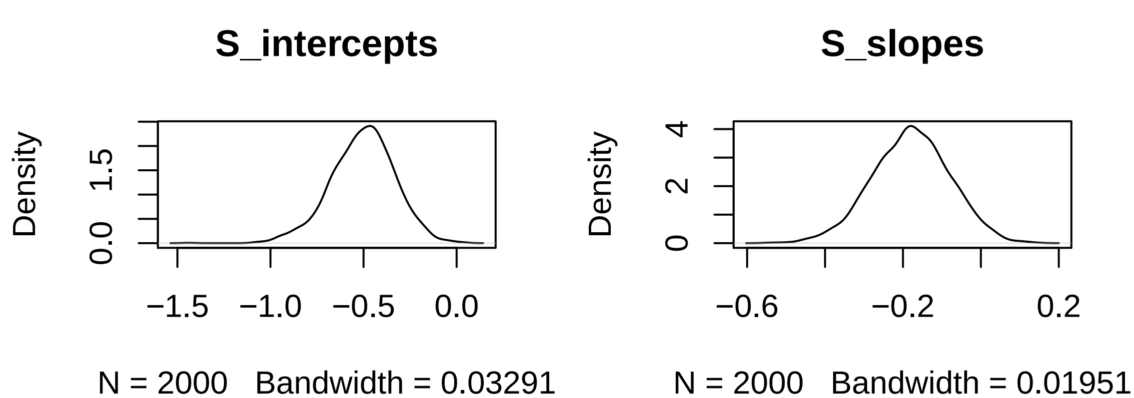
```r
posterior.mode(mcmc(S.modelBV_RR12))
```

```
## S_intercepts     S_slopes
##   -0.4346212   -0.1827471
```

```r
HPDinterval(mcmc(S.modelBV_RR12))
```

```
##                   lower        upper
## S_intercepts -0.8626026 -0.180521187
## S_slopes     -0.3856118  0.006677999
## attr(,"Probability")
## [1] 0.95
```

```r
par(mfrow = c(1,2))
plot(density(S.modelBV_RR12[,1]), main = "S_intercepts")
plot(density(S.modelBV_RR12[,2]), main = "S_slopes")
```



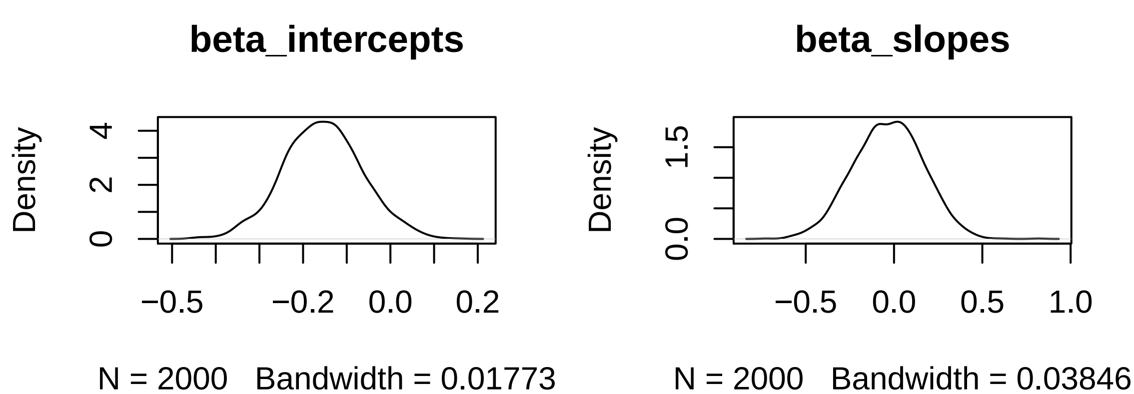
```r
n <- length(modelBV_RR12$VCV[,2])    # sample size
beta_post_RR12 <- matrix(NA, n ,2)

for (i in 1:n) {
  P3 <- matrix(rep(NA, 9), nrow = 3)
  # 3x3 matrix of var-cov for individual X.int, X.slope and LBS
  for (k in 1:9) {P3[k] <- P.modelBV_RR12[i, k] }
  P2 <- P3[1:2, 1:2]    # 2x2 matrix of just trait intercept & slope var-cov
  S <- P3[1:2, 3]    # selection differentials on traits (last column of P3)
  beta_post_RR12[i,] <- solve(P2) %*% S    # selection gradients beta = P^-1 * S
}

colnames(beta_post_RR12) <- c("beta_intercepts", "beta_slopes")
posterior.mode(mcmc(beta_post_RR12))
```

```
## beta_intercepts     beta_slopes
##      -0.1791820       0.0495794
```

```r
HPDinterval(mcmc(beta_post_RR12))
```

```
##                      lower      upper
## beta_intercepts -0.3412324 0.01819105
## beta_slopes     -0.3874528 0.36193375
## attr(,"Probability")
## [1] 0.95
```

```r
par(mfrow = c(1,2))
plot(density(beta_post_RR12[,1]), main = "beta_intercepts")
plot(density(beta_post_RR12[,2]), main = "beta_slopes")
```



The selection differential is "significant" for RN intercepts but not for RN slopes. The selection gradients are not "significant" for any of them. This means that, there is significant total selection (direct + indirect) on intercepts of the RN, but after correcting for the covariance with slopes, there is no direct selection on any of them. There is no significant selection (either direct or indirect) on slopes of the RNs.

With shoot volume

Stack data:

```
# Create a single data-set "data.stack15", with single column at start
# to index observations
data.stack15 <- c()
data.stack15$Obs <- 1:(163 + 1162)
data.stack15$id <- c(as.character(data_5yrs_total$id),as.character(data_5yrs$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values

data.stack15$year <- c(data_5yrs_total$first_yr,
                       data_5yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack15$temp <- c(rep(0, 163), data_5yrs$cmean_4)

# Shoot volume column is only relevant for fitness, but is set to 0 for FFD values
# Using log of mean shoot volume over all years when available, centered
data.stack15$cn_shoot_vol <- c(data_5yrs_total$cn_shoot_vol_mean_log,rep(0, 1162))

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack15$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness_fl), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack15$traits <- c(rep("fitness", 163), rep("FFD", 1162))
data.stack15$variable <- data.stack15$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack15$family <- c(rep("poisson", 163), rep("gaussian", 1162))
data.stack15 <- data.frame(data.stack15)

data.stack15$id <- as.factor(data.stack15$id)
data.stack15$year <- as.factor(data.stack15$year)
head(data.stack15)
```

```
##   Obs      id year temp cn_shoot_vol fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0    1.9543895                16 fitness  fitness poisson
## 2   2 new_100 2006    0    0.3448666                 6 fitness  fitness poisson
## 3   3 new_101 2006    0   -0.1489826                 3 fitness  fitness poisson
## 4   4 new_102 2006    0    0.8572416                 7 fitness  fitness poisson
## 5   5 new_103 2006    0    0.1998053                 4 fitness  fitness poisson
## 6   6 new_104 2006    0   -0.2730598                 3 fitness  fitness poisson
```

```
modelBV_RR15 <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp +  # single fixed effect of temp
                           at.level(variable,"fitness"):cn_shoot_vol,
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
```

```
                           # ^ random intercepts for individual,
                           # and random slopes for temp|id
                           rcov = ~us(at.level(variable, "fitness")):id +
                             # ^ variance between indivduials in fitness
                             # (which is residual variance)
                             us(at.level(variable, "FFD")):Obs,
                             # ^ residual variance within indivduials between years
                           # (labelled by 'Obs')
                           data = data.stack15,
                           prior = priorBiv_RR10,
                           family = NULL, # specified already in the data-set
                           nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                           verbose = F,singular.ok = T)
save(modelBV_RR15,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_R
```

```
kable(summary(modelBV_RR15)$solutions,digits=c(3,3,3,0,3),caption="Fixed effects")
```

Table 16: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	57.400	55.384	59.711	1838	0.000
variablefitness	1.511	1.394	1.634	2000	0.000
at.level(variable, “FFD”):temp	-2.380	-3.914	-0.717	2000	0.007
at.level(variable, “fitness”):cn_shoot_vol	0.360	0.157	0.575	2000	0.001

```
kable(summary(modelBV_RR15)$Gcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 17: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”):at.level(variable, “FFD”).year	25.498	11.906	42.821	2000

```
kable(summary(modelBV_RR15)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 18: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	3.154	1.895	4.525	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	1.028	0.505	1.636	2000
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.319	-0.673	0.009	2000
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	1.028	0.505	1.636	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	0.798	0.365	1.211	2000
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	-0.089	-0.273	0.111	2000
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.319	-0.673	0.009	2000
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	-0.089	-0.273	0.111	2000
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	0.424	0.291	0.568	2143
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	19.302	17.671	21.052	2000

```r
kable(diag(autocorr(modelBV_RR15$Sol)[2, , ]),caption="Autocorrelation")
```

Table 19: Autocorrelation

	x
variableFFD	-0.0141356
variablefitness	-0.0173243
at.level(variable, “FFD”):temp	0.0105147
at.level(variable, “fitness”):cn_shoot_vol	0.0004713

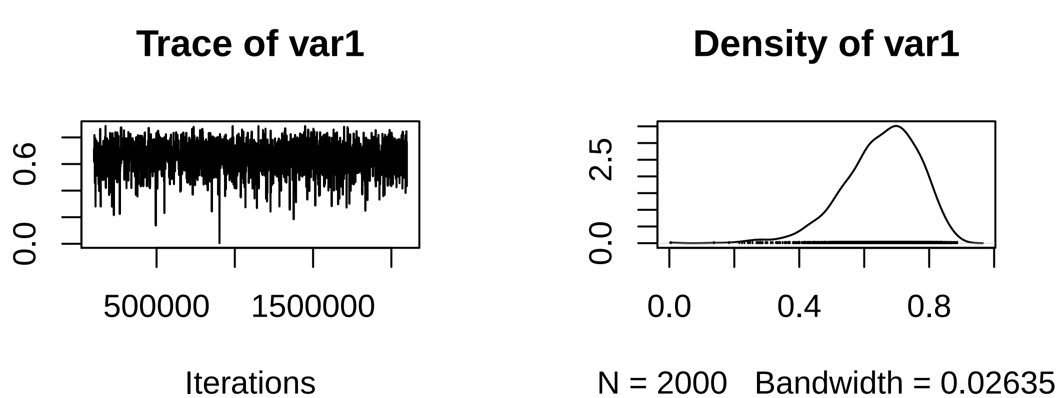
```r
kable(diag(autocorr(modelBV_RR15$VCV)[2, , ]),caption="Autocorrelation")
```

Table 20: Autocorrelation

	x
at.level(variable, “FFD”):at.level(variable, “FFD”).year	0.0044382
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	-0.0291165
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	-0.0068877
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.0236896
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	-0.0068877
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	-0.0229016
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	0.0015387
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.0236896
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	0.0015387
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	-0.0348639
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	0.0061686

Among-individual correlation between intercepts and slopes for FFD:

```r
cor_BV_RR_15_intslope <-
  modelBV_RR15$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"]/
(sqrt(modelBV_RR15$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"])*
sqrt(modelBV_RR15$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_15_intslope)
```



```r
posterior.mode(cor_BV_RR_15_intslope)
```

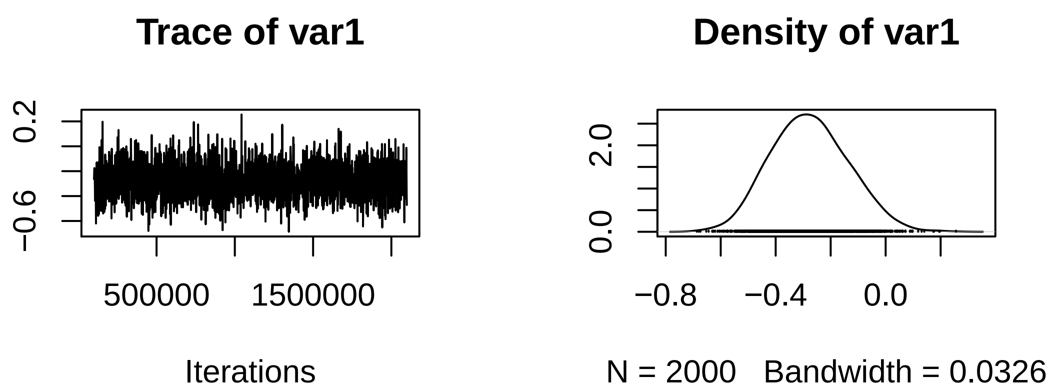
```
##      var1 
## 0.6950366
```

```r
HPDinterval(cor_BV_RR_15_intslope)
```

```
##          lower     upper
## var1 0.4146208 0.8400244
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```r
cor_BV_RR_15_intfit <-
  modelBV_RR15$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\").id"]/
  (sqrt(modelBV_RR15$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR15$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"]))
plot(cor_BV_RR_15_intfit)
```




```r
posterior.mode(cor_BV_RR_15_intfit)
```

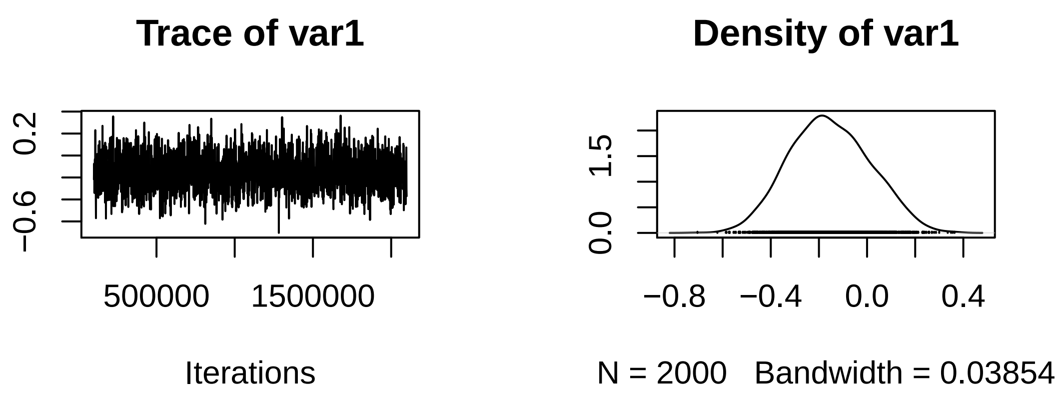
```
##       var1 
## -0.2934346
```

```r
HPDinterval(cor_BV_RR_15_intfit)
```

```
##           lower       upper
## var1 -0.5305822 0.001536857
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between fitness and variation in slopes for FFD:

```
cor_BV_RR_15_slopefit <-
  modelBV_RR15$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]/
  (sqrt(modelBV_RR15$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR15$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])
plot(cor_BV_RR_15_slopefit)
```




```
posterior.mode(cor_BV_RR_15_slopefit)
```

```
##       var1
## -0.1992589
```

```
HPDinterval(cor_BV_RR_15_slopefit)
```

```
##          lower     upper
## var1 -0.473821 0.1593548
## attr(,"Probability")
## [1] 0.95
```

Intercepts and slopes of RNs are positively correlated: Plants that flower earlier on average are also more responsive to temperature. Fitness is not correlated with either the intercept or the slope of the RN: There is no selection on either the intercept or the slope of the RN when including plant size as a condition variable.

```
P.modelBV_RR15 <- modelBV_RR15$VCV[,2:10]
P.modelBV_RR15.mode <- matrix(1:9, nrow = 3)
for (k in 1:9) P.modelBV_RR15.mode[k] <- posterior.mode(P.modelBV_RR15[,k])
P.modelBV_RR15.mode
```

Extract selection coefficients

```
##            [,1]        [,2]        [,3]
## [1,]  3.2252904  0.96413507 -0.21102170
## [2,]  0.9641351  0.72214192 -0.09155512
## [3,] -0.2110217 -0.09155512  0.41434229
```

```
S.modelBV_RR15 <- modelBV_RR15$VCV[, c(4,7)]
S.modelBV_RR15 <- P.modelBV_RR15[, c(3,6)]
colnames(S.modelBV_RR15) <- c("S_intercepts", "S_slopes")
S.modelBV_RR15.mode <- P.modelBV_RR15.mode[1:2, 3]
S.modelBV_RR15.mode
```

```
## [1] -0.21102170 -0.09155512
```

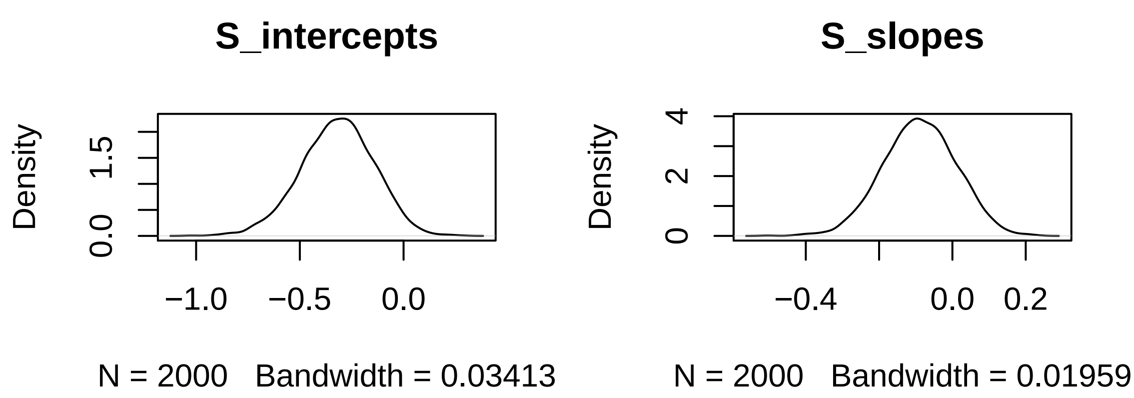
```
posterior.mode(mcmc(S.modelBV_RR15))
```

```
## S_intercepts     S_slopes
##  -0.21102170  -0.09155512
```

```
HPDinterval(mcmc(S.modelBV_RR15))
```

```
##                   lower       upper
## S_intercepts -0.6734277 0.009072126
## S_slopes     -0.2731611 0.111189924
## attr(,"Probability")
## [1] 0.95
```

```
par(mfrow = c(1,2))
plot(density(S.modelBV_RR15[,1]), main = "S_intercepts")
plot(density(S.modelBV_RR15[,2]), main = "S_slopes")
```




```
n <- length(modelBV_RR15$VCV[,2])   # sample size
beta_post_RR15 <- matrix(NA, n ,2)

for (i in 1:n) {
  P3 <- matrix(rep(NA, 9), nrow = 3)
  # 3x3 matrix of var-cov for individual X.int, X.slope and LBS
  for (k in 1:9) {P3[k] <- P.modelBV_RR15[i, k] }
  P2 <- P3[1:2, 1:2]    # 2x2 matrix of just trait intercept & slope var-cov
  S <- P3[1:2, 3]    # selection differentials on traits (last column of P3)
```

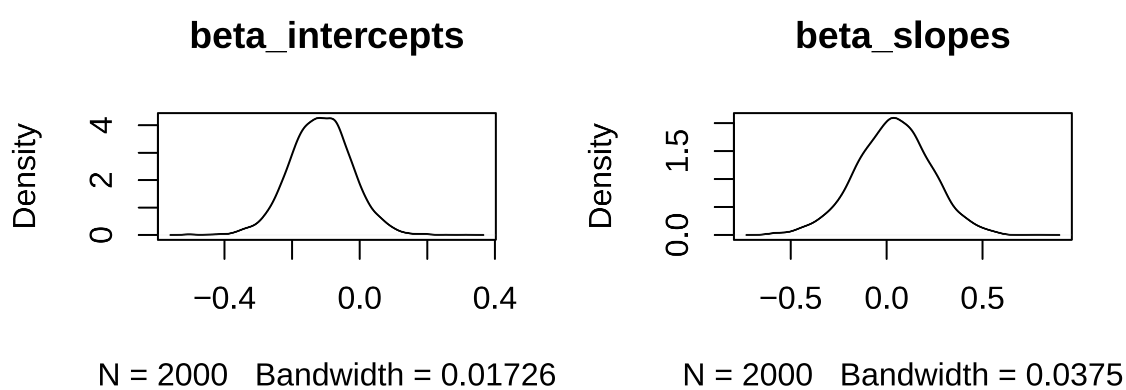
```
  beta_post_RR15[i,] <- solve(P2) %*% S   # selection gradients beta = P^-1 * S
}

colnames(beta_post_RR15) <- c("beta_intercepts", "beta_slopes")
posterior.mode(mcmc(beta_post_RR15))
```

```
## beta_intercepts     beta_slopes
##     -0.07339508      0.02466960
```

```
HPDinterval(mcmc(beta_post_RR15))
```

```
##                      lower      upper
## beta_intercepts -0.2834371 0.06412294
## beta_slopes     -0.3603621 0.40563970
## attr(,"Probability")
## [1] 0.95
```

```
par(mfrow = c(1,2))
plot(density(beta_post_RR15[,1]), main = "beta_intercepts")
plot(density(beta_post_RR15[,2]), main = "beta_slopes")
```



The selection differentials and gradients are not "significant" for either RN intercepts or slopes. This means that there no significant selection (either direct or indirect) on intercepts and slopes of the RNs.

Correlation among size and RN parameters

There is no selection on RN parameters when including plant size as a condition variable. Selection on plasticity might be mediated by the resource state of the plants - this might be indicated by a correlation among plant size and the parameters of the RN. We check this by looking at correlations among plant size (shoot volume) and the BLUPs for intercept and slope of the RN (NOTE: this is maybe not a good use of BLUPs because they do not have associated measures of uncertainty!).

```r
BLUPs<-id_data%>%
  rownames_to_column()%>%
  select(rowname,BLUP_int,BLUP_slope)%>%
  rename(id = rowname)%>%
  right_join(shoot_vol_means)%>%
  mutate(period=ifelse(grepl("old",id),"old","new"))

with(BLUPs,cor.test(shoot_vol_mean,BLUP_int))  # -0.3671608*
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and BLUP_int
## t = -5.0086, df = 161, p-value = 1.429e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4930533 -0.2262075
## sample estimates:
##        cor
## -0.3671608
```

```r
summary(lm(BLUP_int~shoot_vol_mean,BLUPs))
```

```
##
## Call:
## lm(formula = BLUP_int ~ shoot_vol_mean, data = BLUPs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4081 -0.7413 -0.0467  0.7136  3.7588
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.073e-01  1.561e-01   3.891 0.000146 ***
## shoot_vol_mean -3.565e-04  7.118e-05  -5.009 1.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 161 degrees of freedom
## Multiple R-squared:  0.1348, Adjusted R-squared:  0.1294
## F-statistic: 25.09 on 1 and 161 DF,  p-value: 1.429e-06
```

```r
summary(lm(BLUP_int~log(shoot_vol_mean),BLUPs))
```

```
##
## Call:
## lm(formula = BLUP_int ~ log(shoot_vol_mean), data = BLUPs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5628 -0.8008 -0.0355  0.7201  3.9228
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.9139     1.1806   5.009 1.43e-06 ***
## log(shoot_vol_mean)    -0.8174     0.1626  -5.026 1.32e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.254 on 161 degrees of freedom
## Multiple R-squared:  0.1356, Adjusted R-squared:  0.1303
## F-statistic: 25.27 on 1 and 161 DF,  p-value: 1.318e-06
```

```r
with(BLUPs,cor.test(shoot_vol_mean,BLUP_slope))  # -0.3781924*
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and BLUP_slope
## t = -5.1837, df = 161, p-value = 6.444e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5026886 -0.2383273
## sample estimates:
##        cor
## -0.3781924
```

```r
summary(lm(BLUP_slope~shoot_vol_mean,BLUPs))
```

```
##
## Call:
## lm(formula = BLUP_slope ~ shoot_vol_mean, data = BLUPs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43478 -0.31864 -0.01449  0.24080  1.48184
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.460e-01  6.108e-02   4.027 8.67e-05 ***
## shoot_vol_mean -1.444e-04  2.786e-05  -5.184 6.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.491 on 161 degrees of freedom
## Multiple R-squared:  0.143,  Adjusted R-squared:  0.1377
## F-statistic: 26.87 on 1 and 161 DF,  p-value: 6.444e-07
```

```r
summary(lm(BLUP_slope~log(shoot_vol_mean),BLUPs))
```

```
##
## Call:
## lm(formula = BLUP_slope ~ log(shoot_vol_mean), data = BLUPs)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49345 -0.30009 -0.03434  0.27665  1.54701
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.34693    0.46359   5.062 1.12e-06 ***
## log(shoot_vol_mean) -0.32438    0.06385  -5.080 1.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4924 on 161 degrees of freedom
## Multiple R-squared:  0.1382, Adjusted R-squared:  0.1328
## F-statistic: 25.81 on 1 and 161 DF,  p-value: 1.034e-06
```

```
# Old period
with(subset(BLUPs,period=="old"),cor.test(shoot_vol_mean,BLUP_int))  # NS
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and BLUP_int
## t = -0.42432, df = 62, p-value = 0.6728
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2957084  0.1945724
## sample estimates:
##         cor
## -0.05381046
```

```
summary(lm(BLUP_int~shoot_vol_mean,subset(BLUPs,period=="old")))
```

```
##
## Call:
## lm(formula = BLUP_int ~ shoot_vol_mean, data = subset(BLUPs,
##     period == "old"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2929 -0.6201 -0.1504  0.7827  2.4734
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.1692212  0.3413815   0.496    0.622
## shoot_vol_mean -0.0001099  0.0002591  -0.424    0.673
##
## Residual standard error: 1.093 on 62 degrees of freedom
## Multiple R-squared:  0.002896,   Adjusted R-squared:  -0.01319
## F-statistic:  0.18 on 1 and 62 DF,  p-value: 0.6728
```

```
summary(lm(BLUP_int~log(shoot_vol_mean),subset(BLUPs,period=="old")))
```

```
##
## Call:
## lm(formula = BLUP_int ~ log(shoot_vol_mean), data = subset(BLUPs,
##     period == "old"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2986 -0.6085 -0.1652  0.7857  2.4777
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.0894     2.1459   0.508    0.613
## log(shoot_vol_mean)  -0.1504     0.3059  -0.492    0.625
##
## Residual standard error: 1.092 on 62 degrees of freedom
## Multiple R-squared:  0.003884,   Adjusted R-squared:  -0.01218
## F-statistic: 0.2417 on 1 and 62 DF,  p-value: 0.6247
```

```
with(subset(BLUPs,period=="old"),cor.test(shoot_vol_mean,BLUP_slope))  # NS
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and BLUP_slope
## t = -0.29289, df = 62, p-value = 0.7706
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2804188  0.2105614
## sample estimates:
##         cor
## -0.03717178
```

```
summary(lm(BLUP_slope~shoot_vol_mean,subset(BLUPs,period=="old")))
```

```
##
## Call:
## lm(formula = BLUP_slope ~ shoot_vol_mean, data = subset(BLUPs,
##     period == "old"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91284 -0.27166 -0.03862  0.23408  1.13483
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.438e-02  1.230e-01   0.361    0.719
## shoot_vol_mean -2.733e-05  9.331e-05  -0.293    0.771
##
## Residual standard error: 0.3936 on 62 degrees of freedom
## Multiple R-squared:  0.001382,   Adjusted R-squared:  -0.01473
## F-statistic: 0.08579 on 1 and 62 DF,  p-value: 0.7706
```

```r
summary(lm(BLUP_slope~log(shoot_vol_mean),subset(BLUPs,period=="old")))
```

```
##
## Call:
## lm(formula = BLUP_slope ~ log(shoot_vol_mean), data = subset(BLUPs,
##     period == "old"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91184 -0.26971 -0.03659  0.23510  1.13775
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.19822    0.77348   0.256    0.799
## log(shoot_vol_mean) -0.02669    0.11028  -0.242    0.810
##
## Residual standard error: 0.3937 on 62 degrees of freedom
## Multiple R-squared:  0.0009441,  Adjusted R-squared:  -0.01517
## F-statistic: 0.05859 on 1 and 62 DF,  p-value: 0.8095
```

```r
# New period
with(subset(BLUPs,period=="new"),cor.test(shoot_vol_mean,BLUP_int))  # -0.4425795*
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and BLUP_int
## t = -4.8609, df = 97, p-value = 4.499e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5885679 -0.2686388
## sample estimates:
##        cor
## -0.4425795
```

```r
summary(lm(BLUP_int~shoot_vol_mean,subset(BLUPs,period=="new")))
```

```
##
## Call:
## lm(formula = BLUP_int ~ shoot_vol_mean, data = subset(BLUPs,
##     period == "new"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5542 -0.7251 -0.0264  0.6446  3.6527
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.869e-01  2.147e-01   3.665 0.000404 ***
## shoot_vol_mean -4.004e-04  8.237e-05  -4.861  4.5e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.346 on 97 degrees of freedom
## Multiple R-squared:  0.1959, Adjusted R-squared:  0.1876
## F-statistic: 23.63 on 1 and 97 DF,  p-value: 4.499e-06
```

```
summary(lm(BLUP_int~log(shoot_vol_mean),subset(BLUPs,period=="new")))
```

```
##
## Call:
## lm(formula = BLUP_int ~ log(shoot_vol_mean), data = subset(BLUPs,
##     period == "new"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8914 -0.7447 -0.1101  0.6763  3.8327
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           8.2497     1.5228   5.418 4.39e-07 ***
## log(shoot_vol_mean)  -1.1199     0.2054  -5.454 3.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.313 on 97 degrees of freedom
## Multiple R-squared:  0.2347, Adjusted R-squared:  0.2268
## F-statistic: 29.74 on 1 and 97 DF,  p-value: 3.765e-07
```

```
with(subset(BLUPs,period=="new"),cor.test(shoot_vol_mean,BLUP_slope))  # -0.448708*
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and BLUP_slope
## t = -4.945, df = 97, p-value = 3.192e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5935436 -0.2757195
## sample estimates:
##       cor
## -0.448708
```

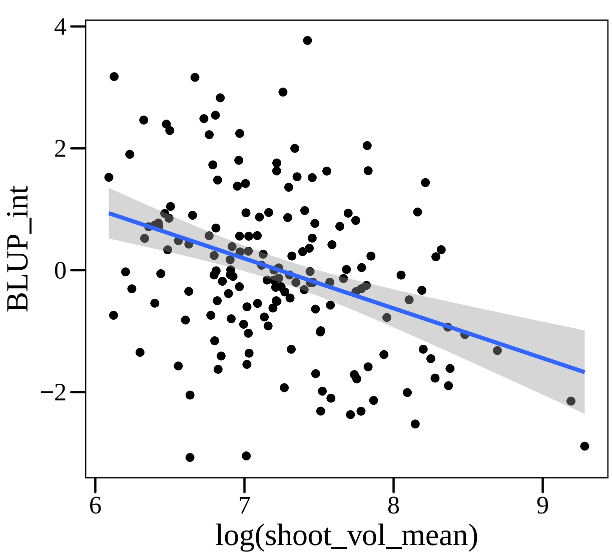
```
summary(lm(BLUP_slope~shoot_vol_mean,subset(BLUPs,period=="new")))
```

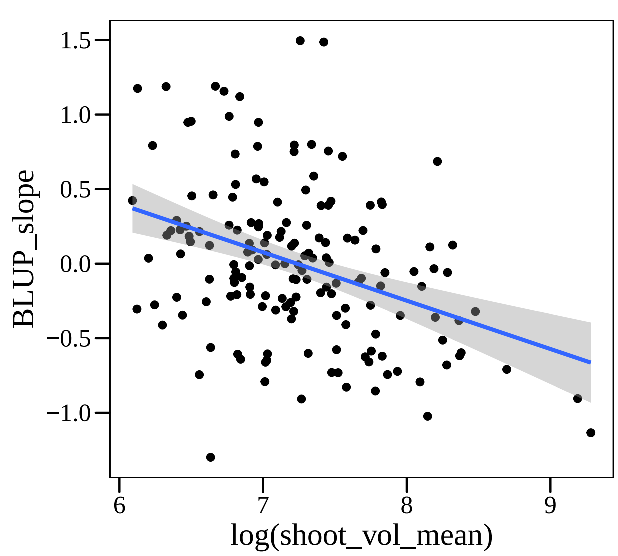
```
##
## Call:
## lm(formula = BLUP_slope ~ shoot_vol_mean, data = subset(BLUPs,
##     period == "new"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49846 -0.29912 -0.03135  0.26451  1.43604
##
```

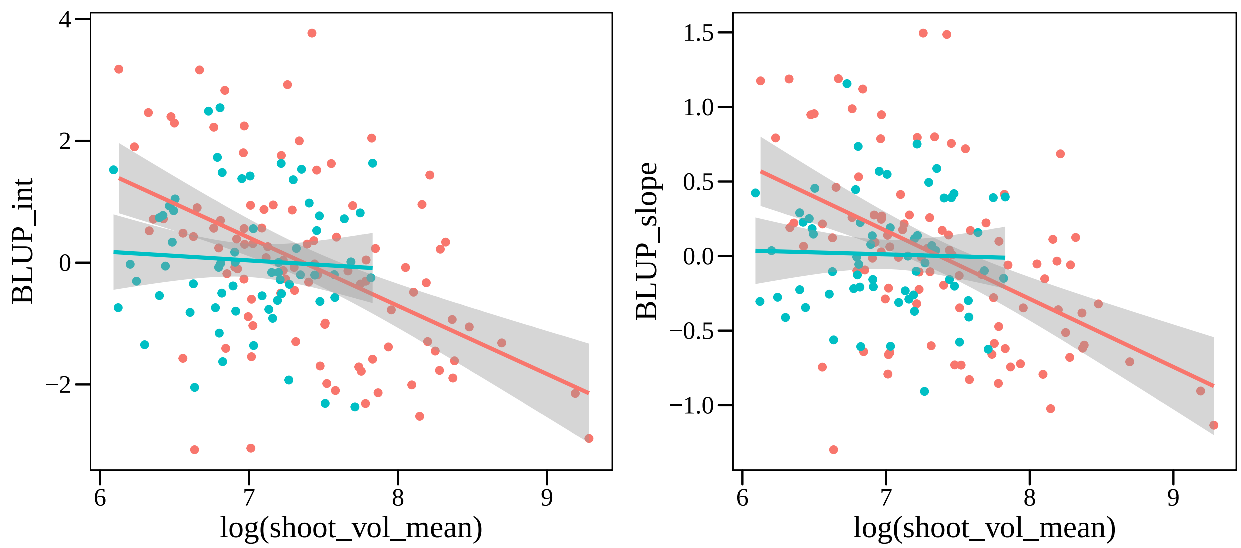
```
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.246e-01  8.644e-02   3.755 0.000296 ***
## shoot_vol_mean -1.640e-04  3.316e-05  -4.945 3.19e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5419 on 97 degrees of freedom
## Multiple R-squared:  0.2013, Adjusted R-squared:  0.1931
## F-statistic: 24.45 on 1 and 97 DF,  p-value: 3.192e-06
```

```
summary(lm(BLUP_slope~log(shoot_vol_mean),subset(BLUPs,period=="new")))
```

```
##
## Call:
## lm(formula = BLUP_slope ~ log(shoot_vol_mean), data = subset(BLUPs,
##     period == "new"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63499 -0.28637 -0.06456  0.27298  1.50968
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.36580    0.61341   5.487 3.26e-07 ***
## log(shoot_vol_mean) -0.45662    0.08272  -5.520 2.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5289 on 97 degrees of freedom
## Multiple R-squared:  0.239,  Adjusted R-squared:  0.2312
## F-statistic: 30.47 on 1 and 97 DF,  p-value: 2.829e-07
```



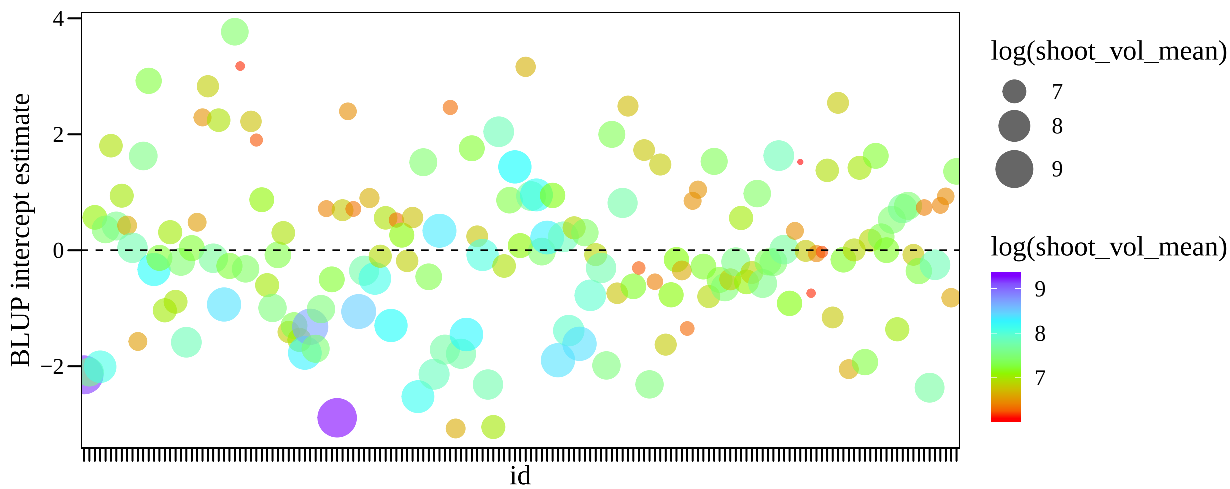




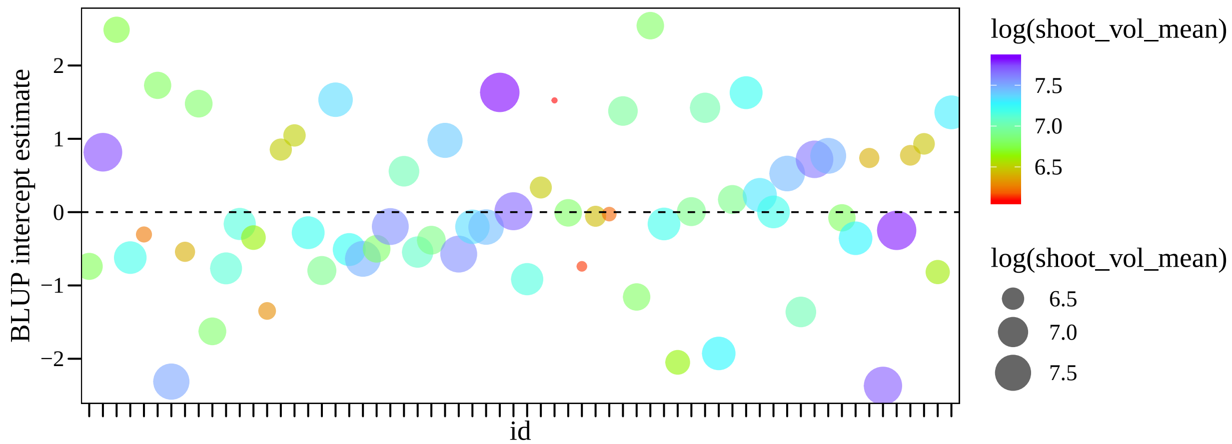
There is a significant negative correlation among size and RN elevation and slope, meaning that larger plants have lower elevations (i.e. flower earlier on average) and slopes (i.e. are more responsive to temperature).

When looking at both periods separately, the correlation is significant only in the "new" period.

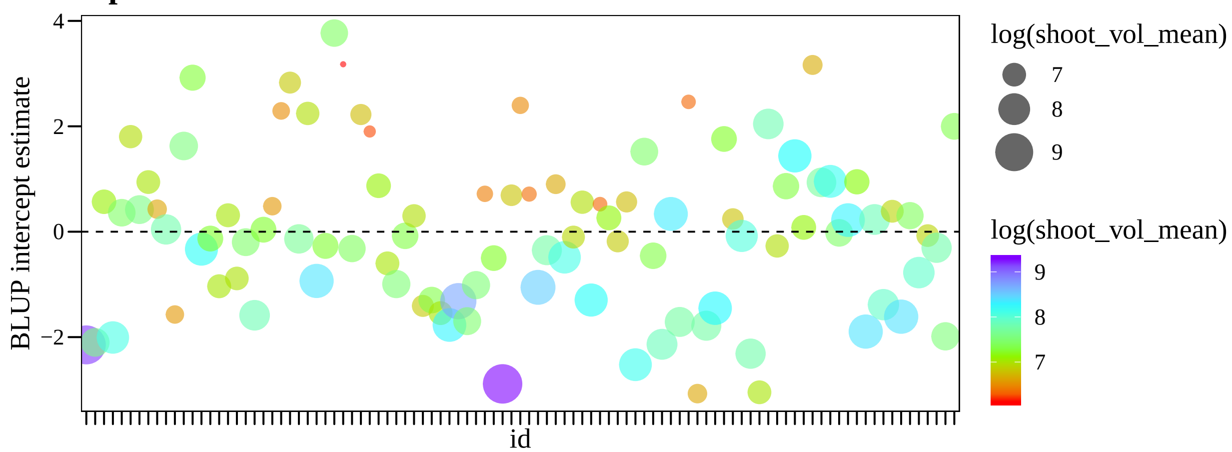
Some graphs for BLUPs showing the size of individuals:

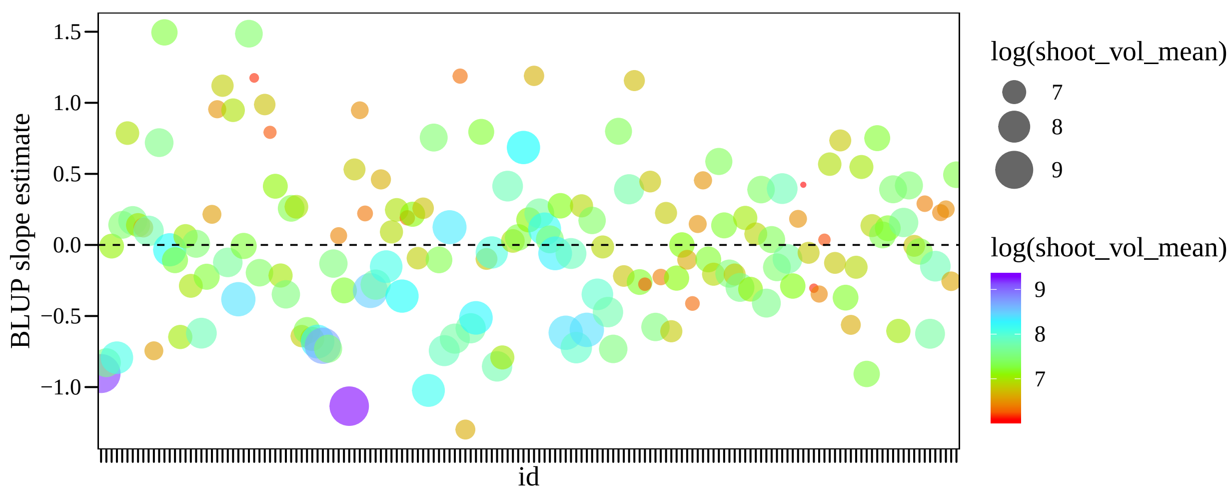


Old period

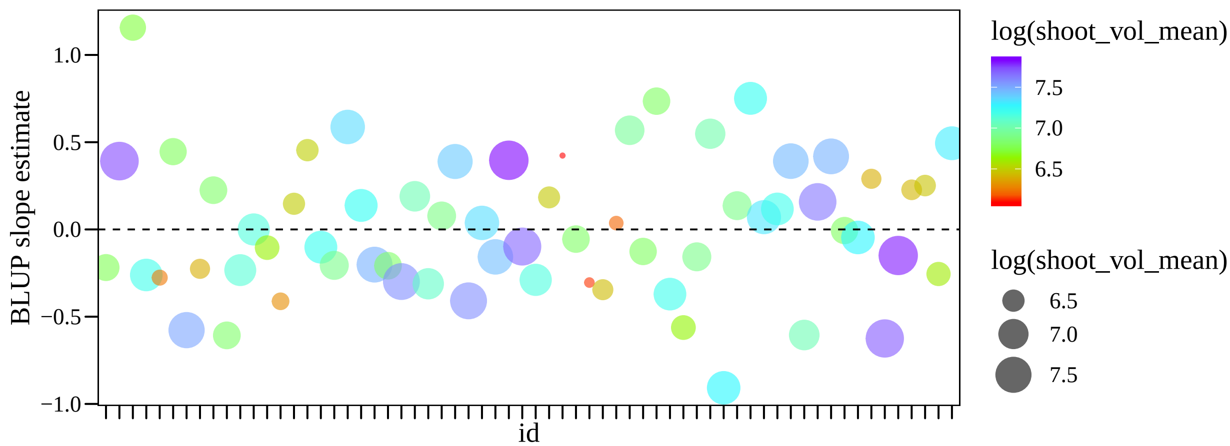


New period

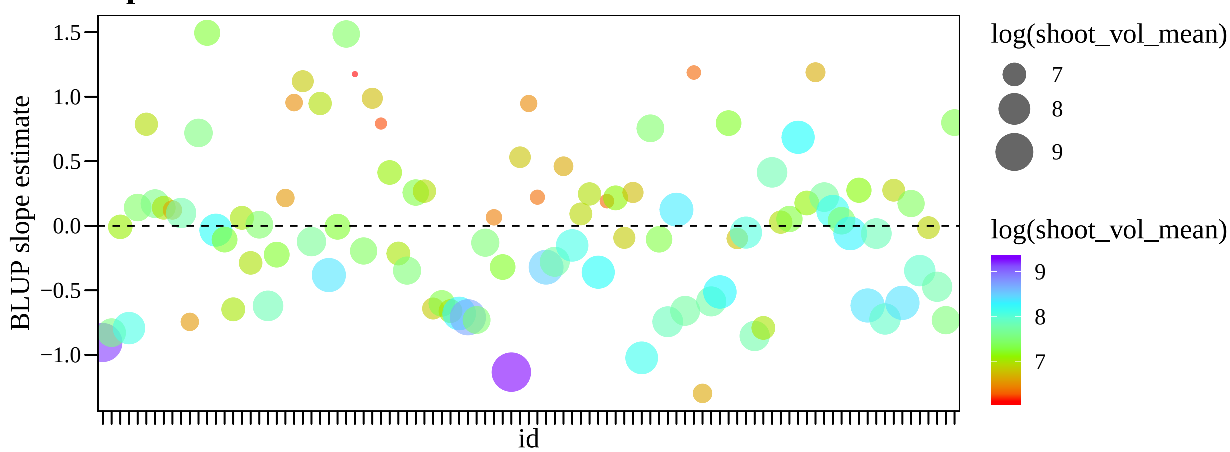


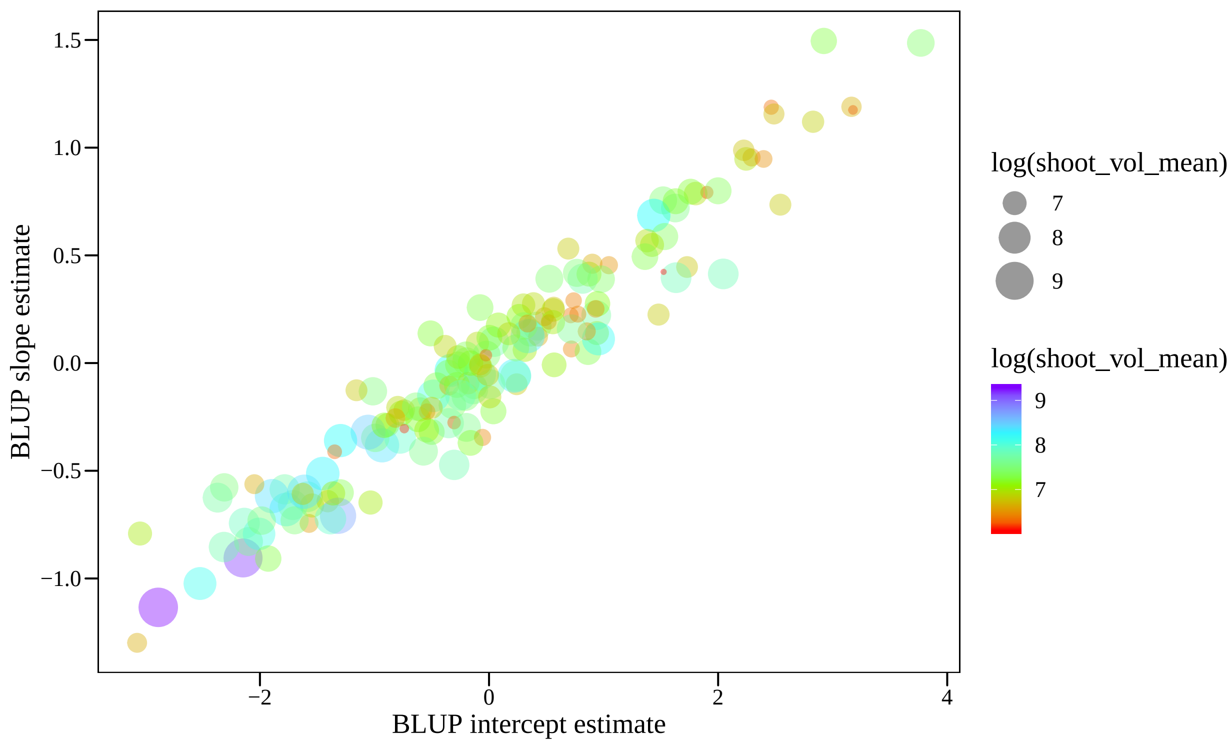


Old period



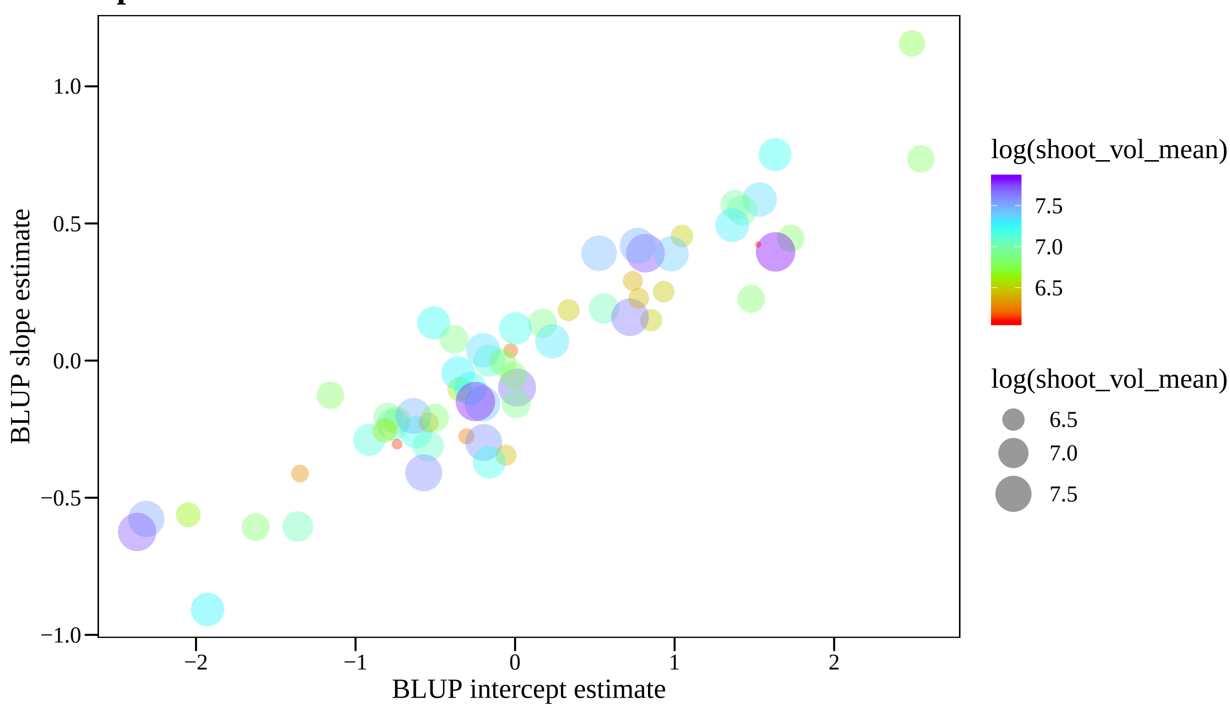
New period





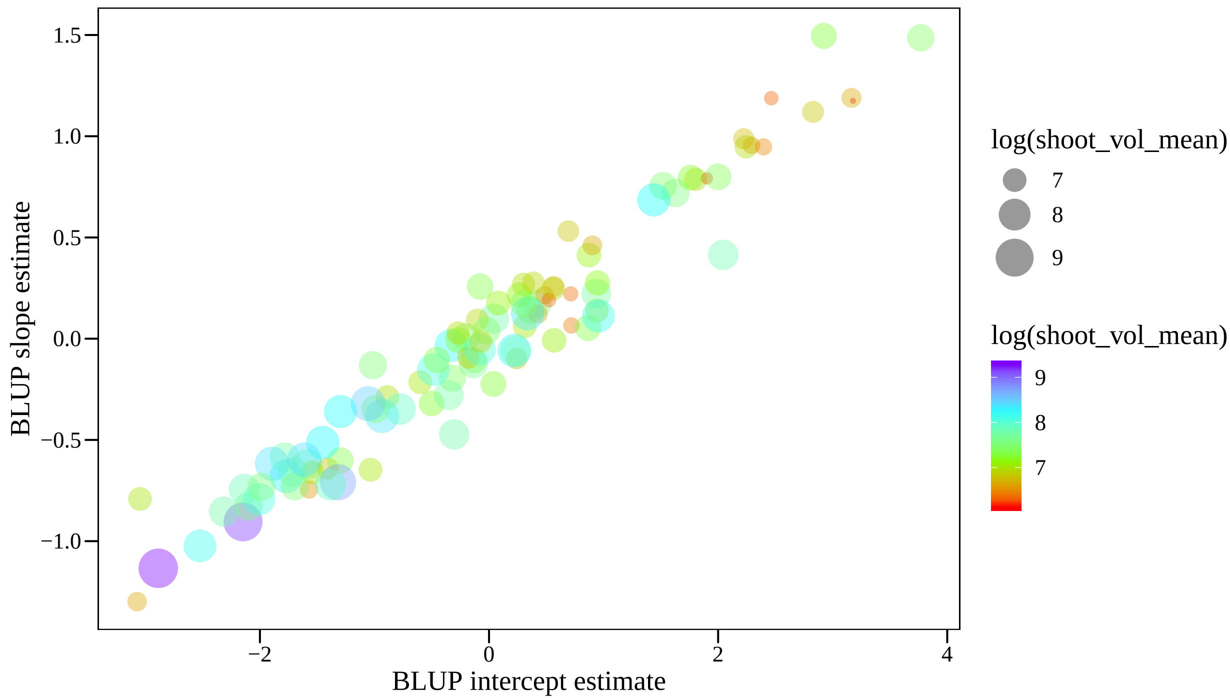
BLUP slope estimate
BLUP intercept estimate
log(shoot_vol_mean)
7
8
9
log(shoot_vol_mean)
9
8
7

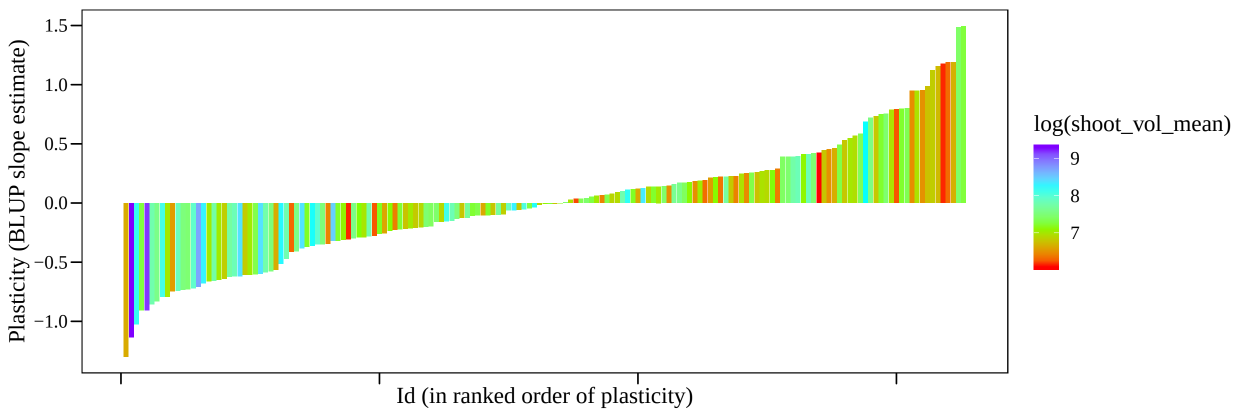
Old period



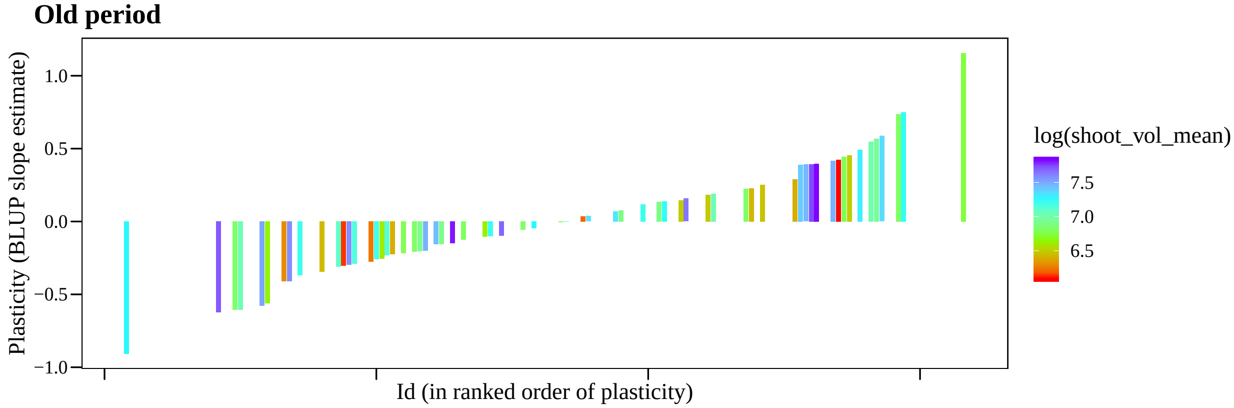
BLUP slope estimate
BLUP intercept estimate
log(shoot_vol_mean)
7.5
7.0
6.5
log(shoot_vol_mean)
6.5
7.0
7.5

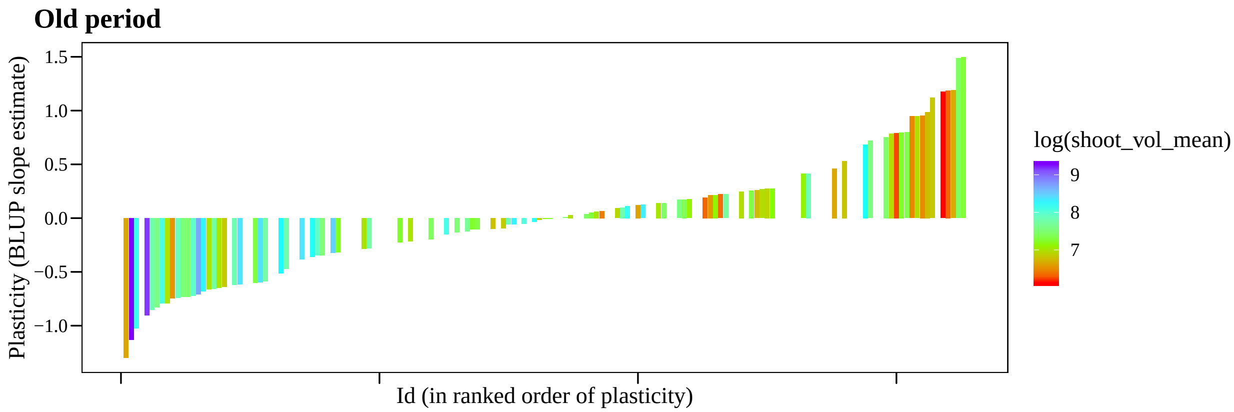
New period





Old period

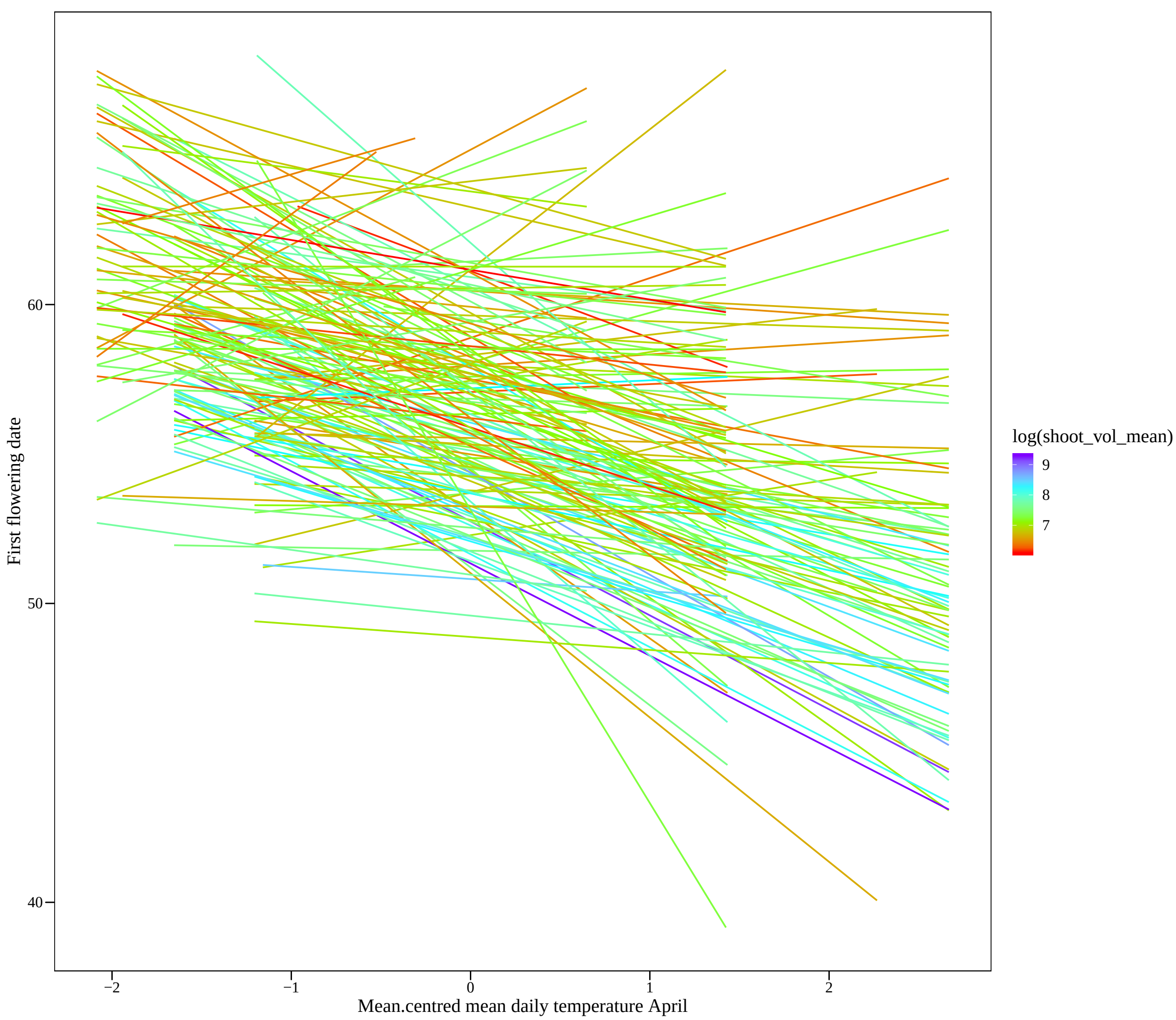




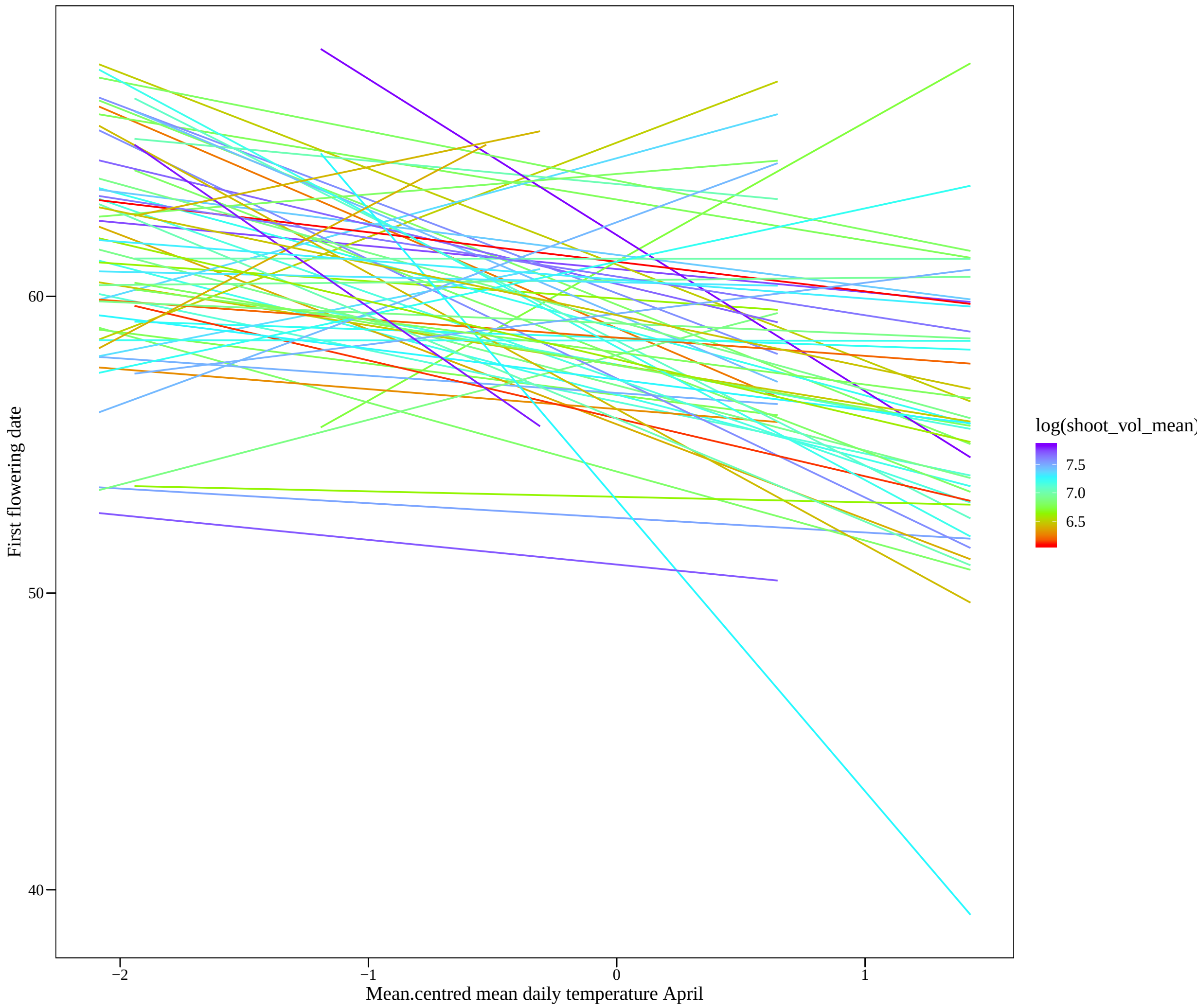
Plant size is significantly correlated with the BLUPs for elevation and slope: this might indicate that selection on RN parameters might be mediated by the resource state of the plants.

When looking at both periods separately, plant size is only significantly correlated with the BLUPs for elevation and slope in the "new" period.

Plot of all the RNs coloured by size:

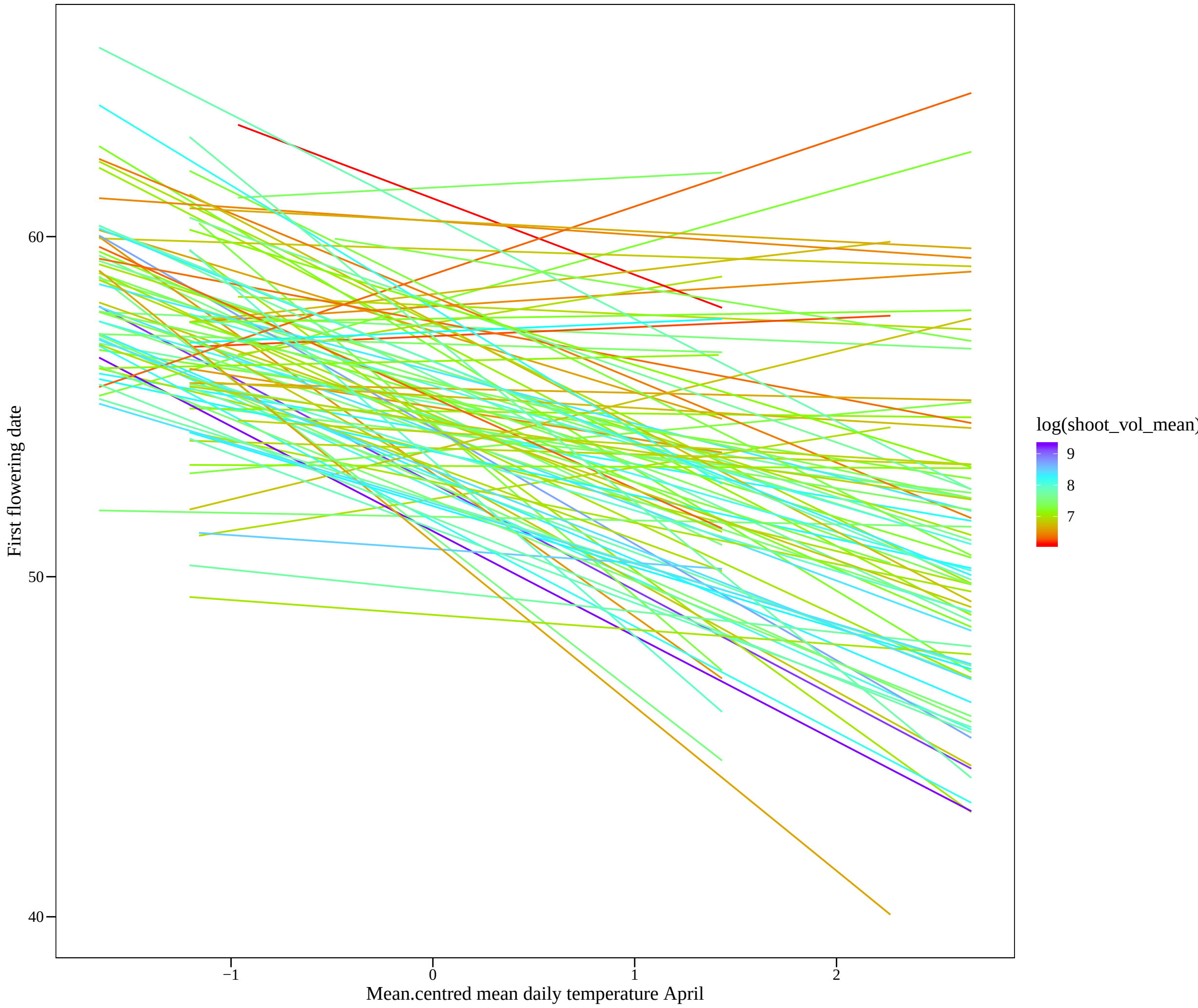


Old period



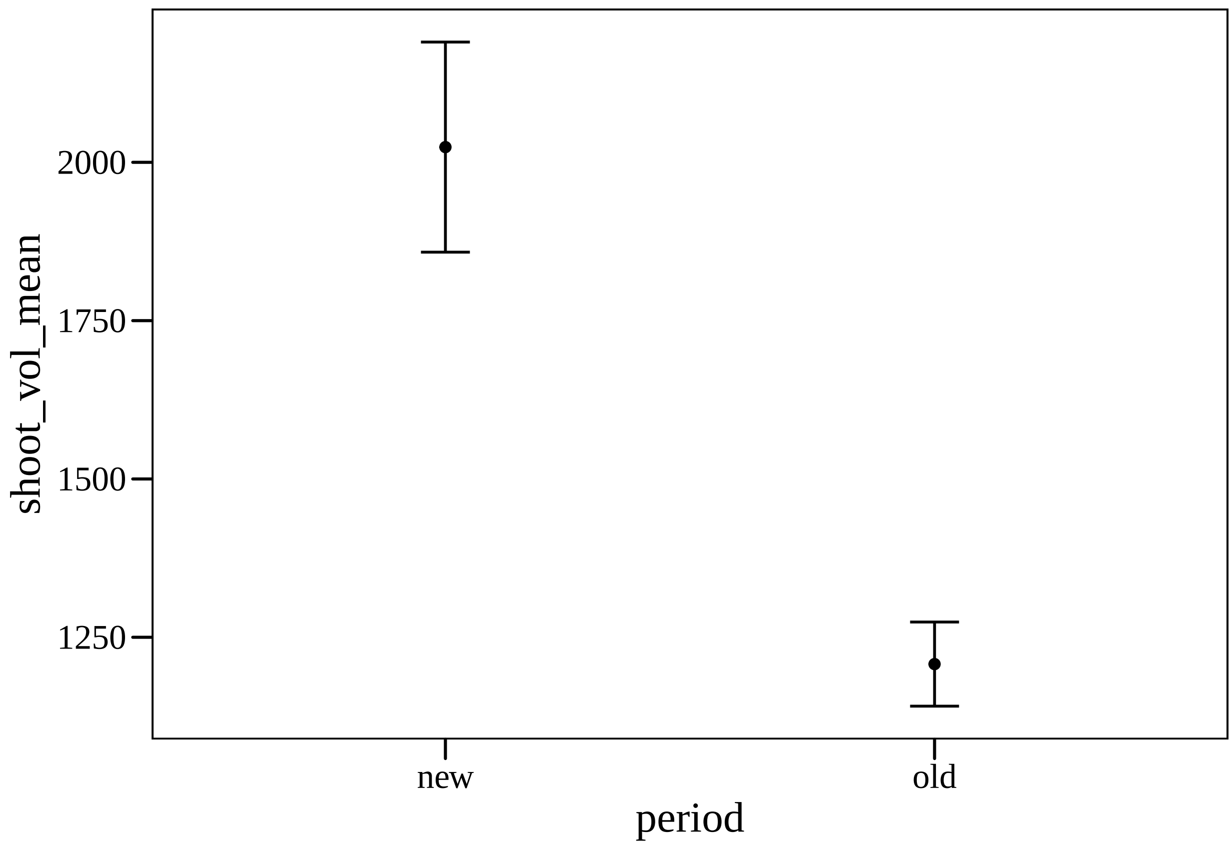
First flowering date
60
50
40
−2
−1
0
1
Mean.centred mean daily temperature April
log(shoot_vol_mean)
7.5
7.0
6.5

New period



60
50
40
First flowering date
−1
0
1
2
Mean.centred mean daily temperature April
log(shoot_vol_mean)
9
8
7

Differences in size between the two periods




```
with(BLUPs,summary(lm(shoot_vol_mean~period)))
```

```
##
## Call:
## lm(formula = shoot_vol_mean ~ period)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1566.2  -758.8  -293.8   374.9  8713.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2024.0      133.7  15.143  < 2e-16 ***
## periodold     -816.3      213.3  -3.827 0.000185 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1330 on 161 degrees of freedom
## Multiple R-squared:  0.08338,    Adjusted R-squared:  0.07769
## F-statistic: 14.65 on 1 and 161 DF,  p-value: 0.0001853
```

```r
with(BLUPs,Anova(lm(shoot_vol_mean~period)))
```

```
## Anova Table (Type II tests)
##
## Response: shoot_vol_mean
##              Sum Sq  Df F value    Pr(>F)
## period     25904207   1  14.646 0.0001853 ***
## Residuals 284756794 161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MCMCglmm models (two periods)

Mean fitness per year of life

With no condition variable

```r
data_5yrs_total<-data_5yrs_total%>%
  mutate(period=ifelse(grepl("old",id),"old","new"))
# Create a single data-set "data.stack.oldA", with single column at start
# to index observations
data.stack.oldA <- c()
data.stack.oldA$Obs <- 1:(64 + 392)
data.stack.oldA$id <- c(as.character(subset(data_5yrs_total,period=="old")$id),
                        as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.oldA$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.oldA$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack.oldA$fitness.FFD.stack <-
  c(round(subset(data_5yrs_total,period=="old")$mean_fitness_life),
    subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.oldA$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.oldA$variable <- data.stack.oldA$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.oldA$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.oldA <- data.frame(data.stack.oldA)

data.stack.oldA$id <- as.factor(data.stack.oldA$id)
```

```r
data.stack.oldA$year <- as.factor(data.stack.oldA$year)
head(data.stack.oldA)
```

Stack data old period

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 2 fitness  fitness poisson
## 2   2 old_133 1987    0                 8 fitness  fitness poisson
## 3   3 old_147 1987    0                 1 fitness  fitness poisson
## 4   4 old_176 1987    0                 1 fitness  fitness poisson
## 5   5 old_182 1987    0                 1 fitness  fitness poisson
## 6   6 old_199 1987    0                 2 fitness  fitness poisson
```

```r
# Create a single data-set "data.stack.newA", with single column at start
# to index observations
data.stack.newA <- c()
data.stack.newA$Obs <- 1:(99 + 770)
data.stack.newA$id <- c(as.character(subset(data_5yrs_total,period=="new")$id),
                        as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.newA$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.newA$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack.newA$fitness.FFD.stack <-
  c(round(subset(data_5yrs_total,period=="new")$mean_fitness_life),
    subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.newA$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.newA$variable <- data.stack.newA$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.newA$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.newA <- data.frame(data.stack.newA)

data.stack.newA$id <- as.factor(data.stack.newA$id)
data.stack.newA$year <- as.factor(data.stack.newA$year)
head(data.stack.newA)
```

Stack data new period

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                14 fitness  fitness poisson
```

```
## 2   2 new_100 2006    0                 4 fitness  fitness poisson
## 3   3 new_101 2006    0                 2 fitness  fitness poisson
## 4   4 new_102 2006    0                 6 fitness  fitness poisson
## 5   5 new_103 2006    0                 4 fitness  fitness poisson
## 6   6 new_104 2006    0                 2 fitness  fitness poisson
```

```
# Scaling factor for MCMCglmm iterations
sc <- 1000 # Increase this parameter for longer runs
modelBV_RR_oldA <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduals in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack.oldA,
                       prior = priorBiv_RR10,
                       family = NULL, # specified already in the data-set
                       nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                       verbose = F,singular.ok = T)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(modelBV_RR_oldA,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
```

Fit model old period

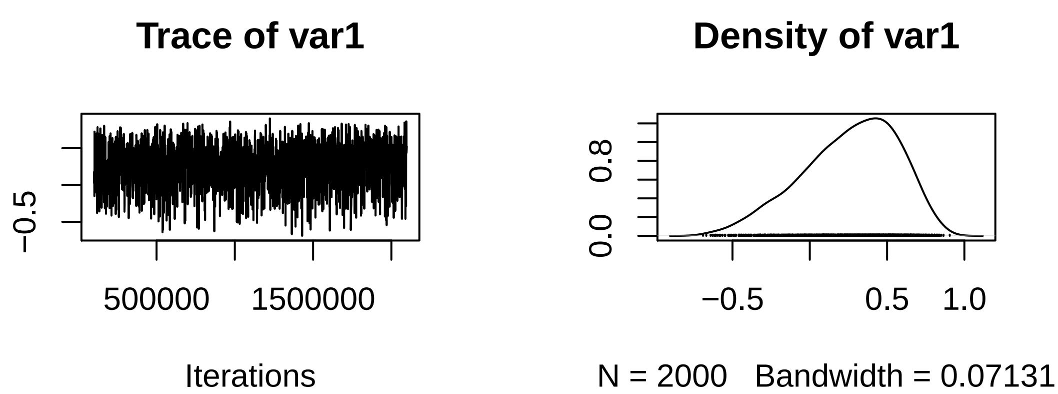
```
modelBV_RR_newA <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduials in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack.newA,
```

```
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
save(modelBV_RR_newA,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
```

Fit model new period

Results old period Among-individual correlation between intercepts and slopes for FFD:

```
cor_BV_RR_oldA_intslope <-
  modelBV_RR_oldA$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"]/
(sqrt(modelBV_RR_oldA$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"])*
sqrt(modelBV_RR_oldA$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_oldA_intslope)
```




```
posterior.mode(cor_BV_RR_oldA_intslope)
```

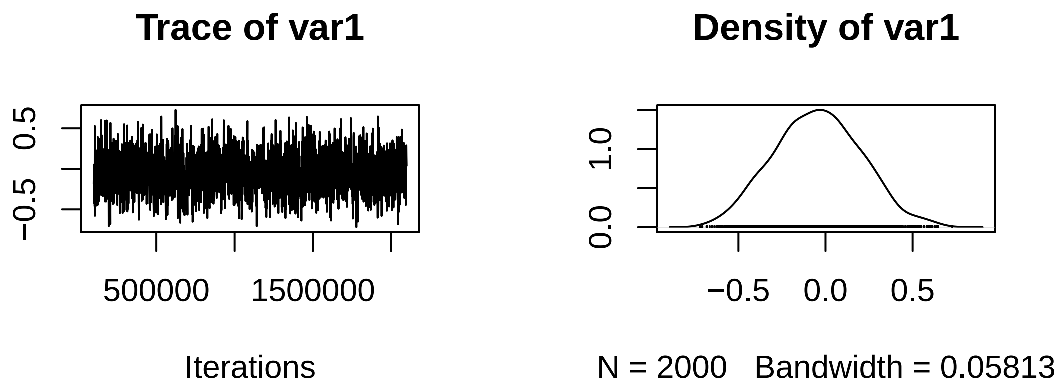
```
##      var1
## 0.5179442
```

```
HPDinterval(cor_BV_RR_oldA_intslope)
```

```
##           lower     upper
## var1 -0.3402055 0.7925493
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
cor_BV_RR_oldA_intfit <-
  modelBV_RR_oldA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\").id"]/
  (sqrt(modelBV_RR_oldA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR_oldA$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"]))
plot(cor_BV_RR_oldA_intfit)
```




```r
posterior.mode(cor_BV_RR_oldA_intfit)
```

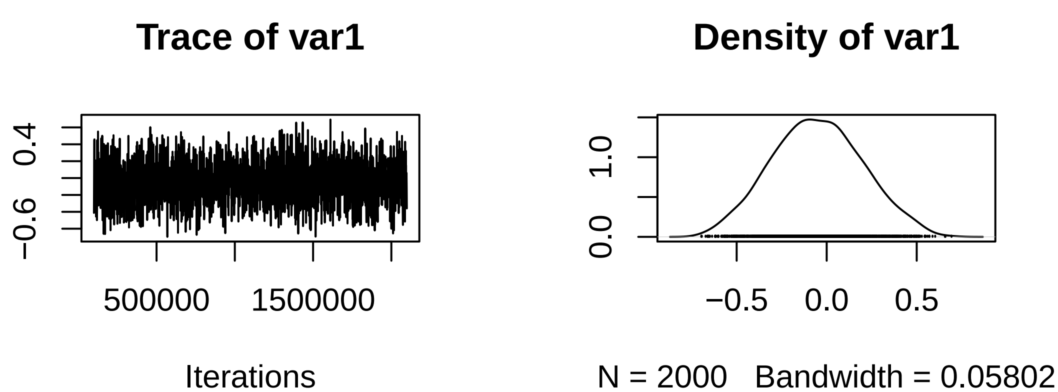
```
##       var1
## 0.05437626
```

```r
HPDinterval(cor_BV_RR_oldA_intfit)
```

```
##           lower    upper
## var1 -0.5518744 0.402904
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between fitness and variation in slopes for FFD:

```r
cor_BV_RR_oldA_slopefit <-
  modelBV_RR_oldA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]/
  (sqrt(modelBV_RR_oldA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR_oldA$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id
plot(cor_BV_RR_oldA_slopefit)
```



```r
posterior.mode(cor_BV_RR_oldA_slopefit)
```

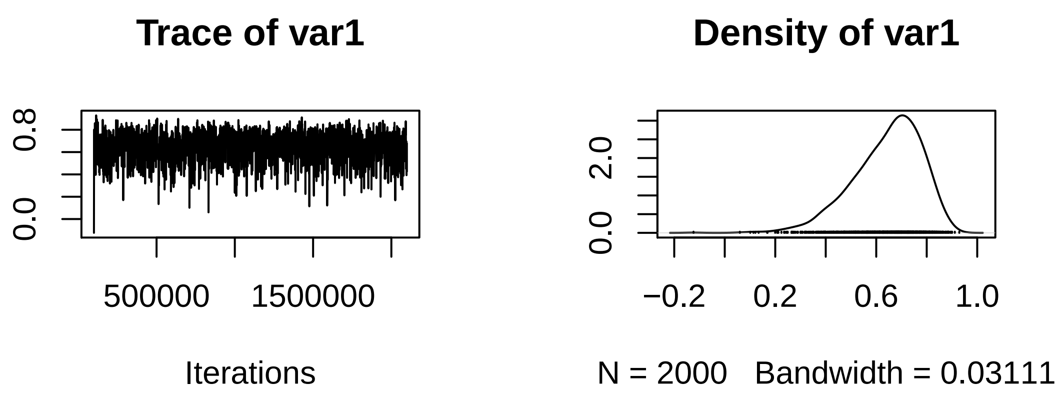
```
##       var1
## 0.04420097
```

```r
HPDinterval(cor_BV_RR_oldA_slopefit)
```

```
##           lower    upper
## var1 -0.5323814 0.440592
## attr(,"Probability")
## [1] 0.95
```

Results new period Among-individual correlation between intercepts and slopes for FFD:

```r
cor_BV_RR_newA_intslope <-
  modelBV_RR_newA$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"]/
(sqrt(modelBV_RR_newA$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"])*
sqrt(modelBV_RR_newA$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_newA_intslope)
```




```r
posterior.mode(cor_BV_RR_newA_intslope)
```

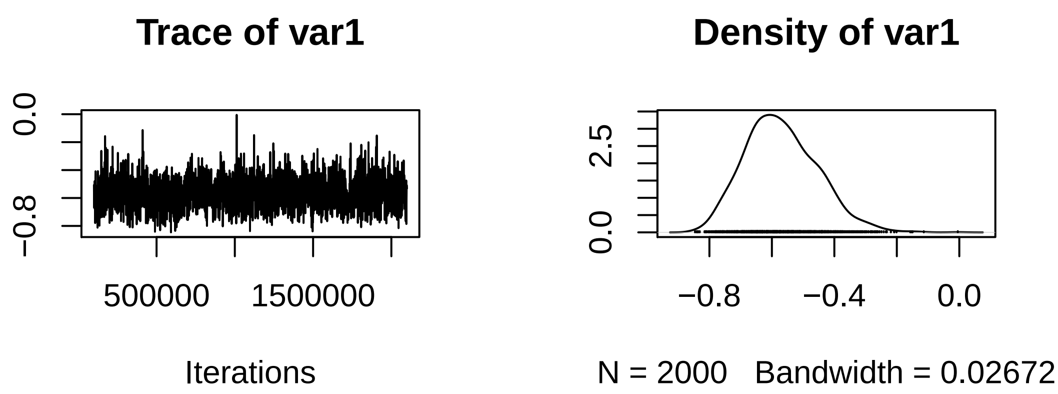
```
##      var1
## 0.7256758
```

```r
HPDinterval(cor_BV_RR_newA_intslope)
```

```
##          lower     upper
## var1 0.3689012 0.8643829
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
cor_BV_RR_newA_intfit <-
  modelBV_RR_newA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\").id"]/
  (sqrt(modelBV_RR_newA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR_newA$VCV[,"at.level(variable, \"FFD\").id:at.level(variable, \"FFD\").id"]))
plot(cor_BV_RR_newA_intfit)
```




```
posterior.mode(cor_BV_RR_newA_intfit)
```

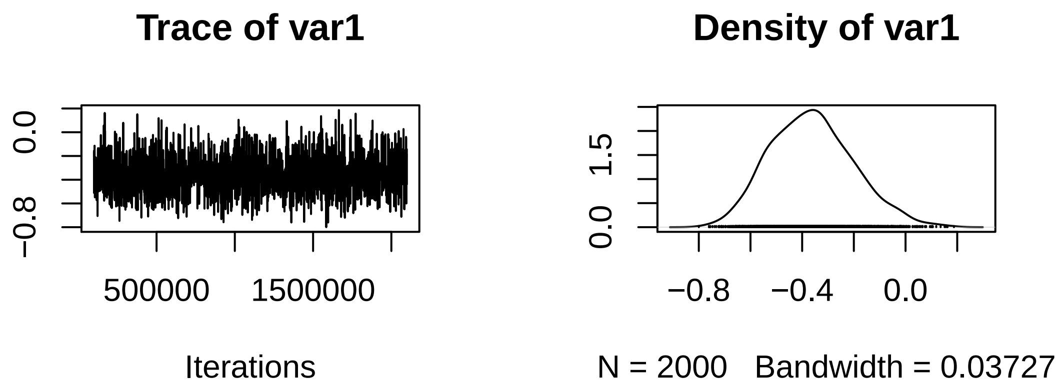
```
##       var1
## -0.6153695
```

```
HPDinterval(cor_BV_RR_newA_intfit)
```

```
##          lower      upper
## var1 -0.782133 -0.3521088
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between fitness and variation in slopes for FFD:

```
cor_BV_RR_newA_slopefit <-
  modelBV_RR_newA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]/
  (sqrt(modelBV_RR_newA$VCV[,"at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])*
     sqrt(modelBV_RR_newA$VCV[,"at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id
plot(cor_BV_RR_newA_slopefit)
```




```r
posterior.mode(cor_BV_RR_newA_slopefit)
```

```
##       var1
## -0.3249277
```

```r
HPDinterval(cor_BV_RR_newA_slopefit)
```

```
##           lower       upper
## var1 -0.6591783 -0.03956803
## attr(,"Probability")
## [1] 0.95
```

Extract selection coefficients old period

Extract selection coefficients new period

MCMCglmm models (yearly)

Data preparation

```r
data_5yrs$period<-ifelse(grepl("old",as.character(data_5yrs$id)),"old","new")
data_5yrs_total$period<-ifelse(grepl("old",as.character(data_5yrs_total$id)),"old","new")
data_5yrs_old<-spread(subset(data_5yrs,period=="old")[c(1,2,6,18)],
                      year, n_intact_seeds)%>%
  rename_at(vars(-id,-period),function(x) paste0("fitness_",x))%>%
  replace(., is.na(.), 0)%>%
  arrange(.,id) # Order by id
data_5yrs_new<-spread(subset(data_5yrs,period=="new")[c(1,2,6,18)],
                      year, n_intact_seeds)%>%
  rename_at(vars(-id,-period),function(x) paste0("fitness_",x))%>%
  replace(., is.na(.), 0)%>%
  arrange(.,id) # Order by id
data_5yrs_old$id<-droplevels(data_5yrs_old$id)
data_5yrs_new$id<-droplevels(data_5yrs_new$id)
length(levels(data_5yrs_old$id)) # 64 id for old period
```

```
## [1] 64
```

```r
length(levels(data_5yrs_new$id)) # 99 ids for new period
```

```
## [1] 99
```

With no condition variable

Stack data

```r
# Create a single data-set "data.stack.1987a", with single column at start to index observations
data.stack.1987a <- c()
data.stack.1987a$Obs <- 1:(64 + 392)
data.stack.1987a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1987a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1987a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1987a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1987),
                                       subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1987a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1987a$variable <- data.stack.1987a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1987a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1987a <- data.frame(data.stack.1987a)

data.stack.1987a$id <- as.factor(data.stack.1987a$id)
data.stack.1987a$year <- as.factor(data.stack.1987a$year)
head(data.stack.1987a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                13 fitness  fitness poisson
## 2   2 old_133 1987    0                20 fitness  fitness poisson
## 3   3 old_147 1987    0                 8 fitness  fitness poisson
## 4   4 old_176 1987    0                10 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                13 fitness  fitness poisson
```

```
##########################################################################################

# Create a single data-set "data.stack.1988a", with single column at start to index observations
data.stack.1988a <- c()
data.stack.1988a$Obs <- 1:(64 + 392)
data.stack.1988a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1988a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1988a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1988a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1988),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1988a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1988a$variable <- data.stack.1988a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1988a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1988a <- data.frame(data.stack.1988a)

data.stack.1988a$id <- as.factor(data.stack.1988a$id)
data.stack.1988a$year <- as.factor(data.stack.1988a$year)
head(data.stack.1988a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
## 2   2 old_133 1987    0                 4 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 2 fitness  fitness poisson
## 6   6 old_199 1987    0                 5 fitness  fitness poisson
```

```
##########################################################################################

# Create a single data-set "data.stack.1989a", with single column at start to index observations
data.stack.1989a <- c()
data.stack.1989a$Obs <- 1:(64 + 392)
data.stack.1989a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1989a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
```

```
                           subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1989a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1989a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1989),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1989a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1989a$variable <- data.stack.1989a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1989a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1989a <- data.frame(data.stack.1989a)

data.stack.1989a$id <- as.factor(data.stack.1989a$id)
data.stack.1989a$year <- as.factor(data.stack.1989a$year)
head(data.stack.1989a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 4 fitness  fitness poisson
## 2   2 old_133 1987    0                22 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 3 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                 0 fitness  fitness poisson
```

```
###########################################################################################

# Create a single data-set "data.stack.1990a", with single column at start to index observations
data.stack.1990a <- c()
data.stack.1990a$Obs <- 1:(64 + 392)
data.stack.1990a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1990a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                           subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1990a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1990a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1990),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
```

```r
data.stack.1990a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1990a$variable <- data.stack.1990a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1990a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1990a <- data.frame(data.stack.1990a)

data.stack.1990a$id <- as.factor(data.stack.1990a$id)
data.stack.1990a$year <- as.factor(data.stack.1990a$year)
head(data.stack.1990a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
## 2   2 old_133 1987    0                22 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                 5 fitness  fitness poisson
```

```r
##########################################################################################

# Create a single data-set "data.stack.1991a", with single column at start to index observations
data.stack.1991a <- c()
data.stack.1991a$Obs <- 1:(64 + 392)
data.stack.1991a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1991a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1991a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1991a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1991),
                                       subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1991a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1991a$variable <- data.stack.1991a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1991a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1991a <- data.frame(data.stack.1991a)

data.stack.1991a$id <- as.factor(data.stack.1991a$id)
data.stack.1991a$year <- as.factor(data.stack.1991a$year)
head(data.stack.1991a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
## 2   2 old_133 1987    0                 0 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 5 fitness  fitness poisson
## 6   6 old_199 1987    0                 0 fitness  fitness poisson
```

```
##########################################################################################

# Create a single data-set "data.stack.1992a", with single column at start to index observations
data.stack.1992a <- c()
data.stack.1992a$Obs <- 1:(64 + 392)
data.stack.1992a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1992a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1992a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1992a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1992),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1992a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1992a$variable <- data.stack.1992a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1992a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1992a <- data.frame(data.stack.1992a)

data.stack.1992a$id <- as.factor(data.stack.1992a$id)
data.stack.1992a$year <- as.factor(data.stack.1992a$year)
head(data.stack.1992a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 3 fitness  fitness poisson
## 2   2 old_133 1987    0                 0 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                 2 fitness  fitness poisson
```

```
##########################################################################################

# Create a single data-set "data.stack.1993a", with single column at start to index observations
```

```r
data.stack.1993a <- c()
data.stack.1993a$Obs <- 1:(64 + 392)
data.stack.1993a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1993a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1993a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1993a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1993),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1993a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1993a$variable <- data.stack.1993a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1993a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1993a <- data.frame(data.stack.1993a)

data.stack.1993a$id <- as.factor(data.stack.1993a$id)
data.stack.1993a$year <- as.factor(data.stack.1993a$year)
head(data.stack.1993a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
## 2   2 old_133 1987    0                 0 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 7 fitness  fitness poisson
## 6   6 old_199 1987    0                 0 fitness  fitness poisson
```

```r
##########################################################################################

# Create a single data-set "data.stack.1994a", with single column at start to index observations
data.stack.1994a <- c()
data.stack.1994a$Obs <- 1:(64 + 392)
data.stack.1994a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1994a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
```

```r
data.stack.1994a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1994a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1994),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1994a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1994a$variable <- data.stack.1994a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1994a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1994a <- data.frame(data.stack.1994a)

data.stack.1994a$id <- as.factor(data.stack.1994a$id)
data.stack.1994a$year <- as.factor(data.stack.1994a$year)
head(data.stack.1994a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
## 2   2 old_133 1987    0                 0 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                 0 fitness  fitness poisson
```

```r
##########################################################################################

# Create a single data-set "data.stack.1995a", with single column at start to index observations
data.stack.1995a <- c()
data.stack.1995a$Obs <- 1:(64 + 392)
data.stack.1995a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1995a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1995a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1995a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1995),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1995a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1995a$variable <- data.stack.1995a$traits
# Fitness will be modelled with an overdispersed Poisson
```

```r
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1995a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1995a <- data.frame(data.stack.1995a)

data.stack.1995a$id <- as.factor(data.stack.1995a$id)
data.stack.1995a$year <- as.factor(data.stack.1995a$year)
head(data.stack.1995a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
## 2   2 old_133 1987    0                14 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                 0 fitness  fitness poisson
```

```r
#################################################################################

# Create a single data-set "data.stack.1996a", with single column at start to index observations
data.stack.1996a <- c()
data.stack.1996a$Obs <- 1:(64 + 392)
data.stack.1996a$id <- c(as.character(data_5yrs_old$id),
                         as.character(subset(data_5yrs,period=="old")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.1996a$year <- c(subset(data_5yrs_total,period=="old")$first_yr,
                          subset(data_5yrs,period=="old")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.1996a$temp <- c(rep(0, 64), subset(data_5yrs,period=="old")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.1996a$fitness.FFD.stack <- c(round(data_5yrs_old$fitness_1996),
                                        subset(data_5yrs,period=="old")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.1996a$traits <- c(rep("fitness", 64), rep("FFD", 392))
data.stack.1996a$variable <- data.stack.1996a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.1996a$family <- c(rep("poisson", 64), rep("gaussian", 392))
data.stack.1996a <- data.frame(data.stack.1996a)

data.stack.1996a$id <- as.factor(data.stack.1996a$id)
data.stack.1996a$year <- as.factor(data.stack.1996a$year)
head(data.stack.1996a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1 old_120 1987    0                 0 fitness  fitness poisson
```

```
## 2   2 old_133 1987    0                 0 fitness  fitness poisson
## 3   3 old_147 1987    0                 0 fitness  fitness poisson
## 4   4 old_176 1987    0                 0 fitness  fitness poisson
## 5   5 old_182 1987    0                 0 fitness  fitness poisson
## 6   6 old_199 1987    0                 0 fitness  fitness poisson
```

```
##########################################################################################

# Create a single data-set "data.stack.2006a", with single column at start to index observations
data.stack.2006a <- c()
data.stack.2006a$Obs <- 1:(99 + 770)
data.stack.2006a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2006a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2006a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2006a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2006),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2006a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2006a$variable <- data.stack.2006a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2006a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2006a <- data.frame(data.stack.2006a)

data.stack.2006a$id <- as.factor(data.stack.2006a$id)
data.stack.2006a$year <- as.factor(data.stack.2006a$year)
head(data.stack.2006a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                43 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                 0 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```
##########################################################################################

# Create a single data-set "data.stack.2007a", with single column at start to index observations
data.stack.2007a <- c()
data.stack.2007a$Obs <- 1:(99 + 770)
```

```r
data.stack.2007a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2007a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2007a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2007a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2007),
                                       subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2007a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2007a$variable <- data.stack.2007a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2007a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2007a <- data.frame(data.stack.2007a)

data.stack.2007a$id <- as.factor(data.stack.2007a$id)
data.stack.2007a$year <- as.factor(data.stack.2007a$year)
head(data.stack.2007a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 7 fitness  fitness poisson
## 2   2 new_100 2006    0                23 fitness  fitness poisson
## 3   3 new_101 2006    0                 2 fitness  fitness poisson
## 4   4 new_102 2006    0                 5 fitness  fitness poisson
## 5   5 new_103 2006    0                 4 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
########################################################################################

# Create a single data-set "data.stack.2008a", with single column at start to index observations
data.stack.2008a <- c()
data.stack.2008a$Obs <- 1:(99 + 770)
data.stack.2008a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2008a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2008a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)
```

```r
# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2008a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2008),
                                       subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2008a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2008a$variable <- data.stack.2008a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2008a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2008a <- data.frame(data.stack.2008a)

data.stack.2008a$id <- as.factor(data.stack.2008a$id)
data.stack.2008a$year <- as.factor(data.stack.2008a$year)
head(data.stack.2008a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                89 fitness  fitness poisson
## 2   2 new_100 2006    0                 2 fitness  fitness poisson
## 3   3 new_101 2006    0                 6 fitness  fitness poisson
## 4   4 new_102 2006    0                31 fitness  fitness poisson
## 5   5 new_103 2006    0                15 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
##########################################################################################

# Create a single data-set "data.stack.2009a", with single column at start to index observations
data.stack.2009a <- c()
data.stack.2009a$Obs <- 1:(99 + 770)
data.stack.2009a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2009a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2009a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2009a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2009),
                                       subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2009a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2009a$variable <- data.stack.2009a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
```

```r
data.stack.2009a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2009a <- data.frame(data.stack.2009a)

data.stack.2009a$id <- as.factor(data.stack.2009a$id)
data.stack.2009a$year <- as.factor(data.stack.2009a$year)
head(data.stack.2009a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                12 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
###########################################################################################

# Create a single data-set "data.stack.2010a", with single column at start to index observations
data.stack.2010a <- c()
data.stack.2010a$Obs <- 1:(99 + 770)
data.stack.2010a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2010a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2010a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2010a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2010),
                                       subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2010a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2010a$variable <- data.stack.2010a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2010a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2010a <- data.frame(data.stack.2010a)

data.stack.2010a$id <- as.factor(data.stack.2010a$id)
data.stack.2010a$year <- as.factor(data.stack.2010a$year)
head(data.stack.2010a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 6 fitness  fitness poisson
```

```
## 4   4 new_102 2006    0                 0 fitness  fitness poisson
## 5   5 new_103 2006    0                 4 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```
###########################################################################################

# Create a single data-set "data.stack.2011a", with single column at start to index observations
data.stack.2011a <- c()
data.stack.2011a$Obs <- 1:(99 + 770)
data.stack.2011a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2011a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2011a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2011a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2011),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2011a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2011a$variable <- data.stack.2011a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2011a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2011a <- data.frame(data.stack.2011a)

data.stack.2011a$id <- as.factor(data.stack.2011a$id)
data.stack.2011a$year <- as.factor(data.stack.2011a$year)
head(data.stack.2011a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                 0 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```
###########################################################################################

# Create a single data-set "data.stack.2012a", with single column at start to index observations
data.stack.2012a <- c()
data.stack.2012a$Obs <- 1:(99 + 770)
data.stack.2012a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
```

```r
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2012a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                           subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2012a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2012a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2012),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2012a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2012a$variable <- data.stack.2012a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2012a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2012a <- data.frame(data.stack.2012a)

data.stack.2012a$id <- as.factor(data.stack.2012a$id)
data.stack.2012a$year <- as.factor(data.stack.2012a$year)
head(data.stack.2012a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 6 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                 7 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
########################################################################################

# Create a single data-set "data.stack.2013a", with single column at start to index observations
data.stack.2013a <- c()
data.stack.2013a$Obs <- 1:(99 + 770)
data.stack.2013a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2013a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                           subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2013a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
```

```r
data.stack.2013a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2013),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2013a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2013a$variable <- data.stack.2013a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2013a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2013a <- data.frame(data.stack.2013a)

data.stack.2013a$id <- as.factor(data.stack.2013a$id)
data.stack.2013a$year <- as.factor(data.stack.2013a$year)
head(data.stack.2013a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                 0 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
#########################################################################################

# Create a single data-set "data.stack.2014a", with single column at start to index observations
data.stack.2014a <- c()
data.stack.2014a$Obs <- 1:(99 + 770)
data.stack.2014a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2014a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2014a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2014a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2014),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2014a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2014a$variable <- data.stack.2014a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2014a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2014a <- data.frame(data.stack.2014a)
```

```r
data.stack.2014a$id <- as.factor(data.stack.2014a$id)
data.stack.2014a$year <- as.factor(data.stack.2014a$year)
head(data.stack.2014a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 8 fitness  fitness poisson
## 3   3 new_101 2006    0                 5 fitness  fitness poisson
## 4   4 new_102 2006    0                 2 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
##########################################################################################

# Create a single data-set "data.stack.2015a", with single column at start to index observations
data.stack.2015a <- c()
data.stack.2015a$Obs <- 1:(99 + 770)
data.stack.2015a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2015a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2015a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2015a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2015),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2015a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2015a$variable <- data.stack.2015a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2015a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2015a <- data.frame(data.stack.2015a)

data.stack.2015a$id <- as.factor(data.stack.2015a$id)
data.stack.2015a$year <- as.factor(data.stack.2015a$year)
head(data.stack.2015a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                 0 fitness  fitness poisson
## 5   5 new_103 2006    0                 8 fitness  fitness poisson
## 6   6 new_104 2006    0                19 fitness  fitness poisson
```

```r
#########################################################################################

# Create a single data-set "data.stack.2016a", with single column at start to index observations
data.stack.2016a <- c()
data.stack.2016a$Obs <- 1:(99 + 770)
data.stack.2016a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2016a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
                          subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2016a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2016a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2016),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2016a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2016a$variable <- data.stack.2016a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2016a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2016a <- data.frame(data.stack.2016a)

data.stack.2016a$id <- as.factor(data.stack.2016a$id)
data.stack.2016a$year <- as.factor(data.stack.2016a$year)
head(data.stack.2016a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                32 fitness  fitness poisson
## 2   2 new_100 2006    0                 8 fitness  fitness poisson
## 3   3 new_101 2006    0                 8 fitness  fitness poisson
## 4   4 new_102 2006    0                10 fitness  fitness poisson
## 5   5 new_103 2006    0                12 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

```r
#########################################################################################

# Create a single data-set "data.stack.2017a", with single column at start to index observations
data.stack.2017a <- c()
data.stack.2017a$Obs <- 1:(99 + 770)
data.stack.2017a$id <- c(as.character(data_5yrs_new$id),
                         as.character(subset(data_5yrs,period=="new")$id))
# ids in alphabetical order

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack.2017a$year <- c(subset(data_5yrs_total,period=="new")$first_yr,
```

```r
                           subset(data_5yrs,period=="new")$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack.2017a$temp <- c(rep(0, 99), subset(data_5yrs,period=="new")$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES) for the particular year,
# then FFD values:
data.stack.2017a$fitness.FFD.stack <- c(round(data_5yrs_new$fitness_2017),
                                        subset(data_5yrs,period=="new")$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack.2017a$traits <- c(rep("fitness", 99), rep("FFD", 770))
data.stack.2017a$variable <- data.stack.2017a$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack.2017a$family <- c(rep("poisson", 99), rep("gaussian", 770))
data.stack.2017a <- data.frame(data.stack.2017a)

data.stack.2017a$id <- as.factor(data.stack.2017a$id)
data.stack.2017a$year <- as.factor(data.stack.2017a$year)
head(data.stack.2017a)
```

```
##   Obs      id year temp fitness.FFD.stack  traits variable  family
## 1   1  new_10 2006    0                 0 fitness  fitness poisson
## 2   2 new_100 2006    0                 0 fitness  fitness poisson
## 3   3 new_101 2006    0                 0 fitness  fitness poisson
## 4   4 new_102 2006    0                 0 fitness  fitness poisson
## 5   5 new_103 2006    0                 0 fitness  fitness poisson
## 6   6 new_104 2006    0                 0 fitness  fitness poisson
```

Run models

```r
modelBV_RR_1987a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduals in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack.1987a,
                       prior = priorBiv_RR10,
                       family = NULL, # specified already in the data-set
                       nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
```

```r
                            verbose = F,singular.ok = T)
modelBV_RR_1988a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                              # ^ means for each variable (and no overall mean (hence "-1"))
                              at.level(variable, "FFD"):temp,  # single fixed effect of temp
                              random = ~us(at.level(variable, "FFD")):year +
                                us(at.level(variable, "FFD") +
                                     at.level(variable,"FFD"):temp):id,
                            # ^ random intercepts for individual,
                            # and random slopes for temp|id
                            rcov = ~us(at.level(variable, "fitness")):id +
                              # ^ variance between indivdiuals in fitness
                              # (which is residual variance)
                              us(at.level(variable, "FFD")):Obs,
                              # ^ residual variance within indivdiuals between years
                            # (labelled by 'Obs')
                            data = data.stack.1988a,
                            prior = priorBiv_RR10,
                            family = NULL, # specified already in the data-set
                            nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                            verbose = F,singular.ok = T)
modelBV_RR_1989a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                              # ^ means for each variable (and no overall mean (hence "-1"))
                              at.level(variable, "FFD"):temp,  # single fixed effect of temp
                              random = ~us(at.level(variable, "FFD")):year +
                                us(at.level(variable, "FFD") +
                                     at.level(variable,"FFD"):temp):id,
                            # ^ random intercepts for individual,
                            # and random slopes for temp|id
                            rcov = ~us(at.level(variable, "fitness")):id +
                              # ^ variance between indivdiuals in fitness
                              # (which is residual variance)
                              us(at.level(variable, "FFD")):Obs,
                              # ^ residual variance within indivdiuals between years
                            # (labelled by 'Obs')
                            data = data.stack.1989a,
                            prior = priorBiv_RR10,
                            family = NULL, # specified already in the data-set
                            nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                            verbose = F,singular.ok = T)
modelBV_RR_1990a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                              # ^ means for each variable (and no overall mean (hence "-1"))
                              at.level(variable, "FFD"):temp,  # single fixed effect of temp
                              random = ~us(at.level(variable, "FFD")):year +
                                us(at.level(variable, "FFD") +
                                     at.level(variable,"FFD"):temp):id,
                            # ^ random intercepts for individual,
                            # and random slopes for temp|id
                            rcov = ~us(at.level(variable, "fitness")):id +
                              # ^ variance between indivdiuals in fitness
                              # (which is residual variance)
                              us(at.level(variable, "FFD")):Obs,
                              # ^ residual variance within indivdiuals between years
                            # (labelled by 'Obs')
```

```
                        data = data.stack.1990a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_1991a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.1991a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_1992a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.1992a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_1993a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
```

```
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.1993a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_1994a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.1994a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_1995a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.1995a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_1996a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
```

```
                         # ^ random intercepts for individual,
                         # and random slopes for temp|id
                         rcov = ~us(at.level(variable, "fitness")):id +
                           # ^ variance between indivduials in fitness
                           # (which is residual variance)
                           us(at.level(variable, "FFD")):Obs,
                           # ^ residual variance within indivduials between years
                         # (labelled by 'Obs')
                         data = data.stack.1996a,
                         prior = priorBiv_RR10,
                         family = NULL, # specified already in the data-set
                         nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                         verbose = F,singular.ok = T)
modelBV_RR_2006a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                           # ^ means for each variable (and no overall mean (hence "-1"))
                           at.level(variable, "FFD"):temp,  # single fixed effect of temp
                           random = ~us(at.level(variable, "FFD")):year +
                             us(at.level(variable, "FFD") +
                                  at.level(variable,"FFD"):temp):id,
                         # ^ random intercepts for individual,
                         # and random slopes for temp|id
                         rcov = ~us(at.level(variable, "fitness")):id +
                           # ^ variance between indivduials in fitness
                           # (which is residual variance)
                           us(at.level(variable, "FFD")):Obs,
                           # ^ residual variance within indivduials between years
                         # (labelled by 'Obs')
                         data = data.stack.2006a,
                         prior = priorBiv_RR10,
                         family = NULL, # specified already in the data-set
                         nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                         verbose = F,singular.ok = T)
modelBV_RR_2007a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                           # ^ means for each variable (and no overall mean (hence "-1"))
                           at.level(variable, "FFD"):temp,  # single fixed effect of temp
                           random = ~us(at.level(variable, "FFD")):year +
                             us(at.level(variable, "FFD") +
                                  at.level(variable,"FFD"):temp):id,
                         # ^ random intercepts for individual,
                         # and random slopes for temp|id
                         rcov = ~us(at.level(variable, "fitness")):id +
                           # ^ variance between indivduials in fitness
                           # (which is residual variance)
                           us(at.level(variable, "FFD")):Obs,
                           # ^ residual variance within indivduials between years
                         # (labelled by 'Obs')
                         data = data.stack.2007a,
                         prior = priorBiv_RR10,
                         family = NULL, # specified already in the data-set
                         nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                         verbose = F,singular.ok = T)
modelBV_RR_2008a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                           # ^ means for each variable (and no overall mean (hence "-1"))
```

```
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduials in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack.2008a,
                       prior = priorBiv_RR10,
                       family = NULL, # specified already in the data-set
                       nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                       verbose = F,singular.ok = T)
modelBV_RR_2009a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduials in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack.2009a,
                       prior = priorBiv_RR10,
                       family = NULL, # specified already in the data-set
                       nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                       verbose = F,singular.ok = T)
modelBV_RR_2010a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                         # ^ means for each variable (and no overall mean (hence "-1"))
                         at.level(variable, "FFD"):temp,  # single fixed effect of temp
                         random = ~us(at.level(variable, "FFD")):year +
                           us(at.level(variable, "FFD") +
                                at.level(variable,"FFD"):temp):id,
                       # ^ random intercepts for individual,
                       # and random slopes for temp|id
                       rcov = ~us(at.level(variable, "fitness")):id +
                         # ^ variance between indivduials in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduials between years
                       # (labelled by 'Obs')
                       data = data.stack.2010a,
                       prior = priorBiv_RR10,
                       family = NULL, # specified already in the data-set
```

```
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_2011a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.2011a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_2012a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.2012a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_2013a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
```

```
                        # (labelled by 'Obs')
                        data = data.stack.2013a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_2014a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.2014a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_2015a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
                          # ^ variance between indivduials in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within indivduials between years
                        # (labelled by 'Obs')
                        data = data.stack.2015a,
                        prior = priorBiv_RR10,
                        family = NULL, # specified already in the data-set
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                        verbose = F,singular.ok = T)
modelBV_RR_2016a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                          # ^ means for each variable (and no overall mean (hence "-1"))
                          at.level(variable, "FFD"):temp,  # single fixed effect of temp
                          random = ~us(at.level(variable, "FFD")):year +
                            us(at.level(variable, "FFD") +
                                 at.level(variable,"FFD"):temp):id,
                        # ^ random intercepts for individual,
                        # and random slopes for temp|id
                        rcov = ~us(at.level(variable, "fitness")):id +
```

```
                        # ^ variance between indivduals in fitness
                        # (which is residual variance)
                        us(at.level(variable, "FFD")):Obs,
                        # ^ residual variance within indivduials between years
                      # (labelled by 'Obs')
                      data = data.stack.2016a,
                      prior = priorBiv_RR10,
                      family = NULL, # specified already in the data-set
                      nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                      verbose = F,singular.ok = T)
modelBV_RR_2017a <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                        # ^ means for each variable (and no overall mean (hence "-1"))
                        at.level(variable, "FFD"):temp,  # single fixed effect of temp
                        random = ~us(at.level(variable, "FFD")):year +
                          us(at.level(variable, "FFD") +
                               at.level(variable,"FFD"):temp):id,
                      # ^ random intercepts for individual,
                      # and random slopes for temp|id
                      rcov = ~us(at.level(variable, "fitness")):id +
                        # ^ variance between indivduals in fitness
                        # (which is residual variance)
                        us(at.level(variable, "FFD")):Obs,
                        # ^ residual variance within indivduials between years
                      # (labelled by 'Obs')
                      data = data.stack.2017a,
                      prior = priorBiv_RR10,
                      family = NULL, # specified already in the data-set
                      nitt = 2100 * sc, thin = sc, burnin = 100 * sc,
                      verbose = F,singular.ok = T)
save(modelBV_RR_1987a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1988a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1989a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1990a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1991a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1992a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1993a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1994a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1995a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_1996a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2006a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2007a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2008a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2009a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2010a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2011a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2012a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2013a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2014a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2015a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2016a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
save(modelBV_RR_2017a,file="C:/Users/avald/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelB
```

Results

```r
posterior.mode(cor_BV_RR_1987a_intslope)
```

Among-individual correlation between intercepts and slopes for FFD:

```
##      var1
## 0.2698776
```

```r
HPDinterval(cor_BV_RR_1987a_intslope)
```

```
##           lower     upper
## var1 -0.3615968 0.7664261
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1988a_intslope)
```

```
##      var1
## 0.2952414
```

```r
HPDinterval(cor_BV_RR_1988a_intslope)
```

```
##           lower     upper
## var1 -0.3541476 0.7754943
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1989a_intslope)
```

```
##      var1
## 0.5029786
```

```r
HPDinterval(cor_BV_RR_1989a_intslope)
```

```
##           lower     upper
## var1 -0.2213016 0.8098679
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1990a_intslope)
```

```
##      var1
## 0.4041568
```

```r
HPDinterval(cor_BV_RR_1990a_intslope)
```

```
##            lower     upper
## var1 -0.2221191 0.8082528
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1991a_intslope)
```

```
##      var1
## 0.3957279
```

```r
HPDinterval(cor_BV_RR_1991a_intslope)
```

```
##            lower     upper
## var1 -0.3150886 0.7685974
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1992a_intslope)
```

```
##      var1
## 0.4778802
```

```r
HPDinterval(cor_BV_RR_1992a_intslope)
```

```
##           lower     upper
## var1 -0.336157 0.7744917
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1993a_intslope)
```

```
##     var1
## 0.345396
```

```r
HPDinterval(cor_BV_RR_1993a_intslope)
```

```
##            lower     upper
## var1 -0.3918171 0.7514067
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1994a_intslope)
```

```
##      var1
## 0.2682107
```

```r
HPDinterval(cor_BV_RR_1994a_intslope)
```

```
##           lower     upper
## var1 -0.3260778 0.7394138
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1995a_intslope)
```

```
##      var1
## 0.2937712
```

```r
HPDinterval(cor_BV_RR_1995a_intslope)
```

```
##           lower     upper
## var1 -0.2991193 0.7913186
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1996a_intslope)
```

```
##      var1
## 0.3285952
```

```r
HPDinterval(cor_BV_RR_1996a_intslope)
```

```
##           lower     upper
## var1 -0.3287141 0.8098586
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2006a_intslope)
```

```
##     var1
## 0.708225
```

```r
HPDinterval(cor_BV_RR_2006a_intslope)
```

```
##          lower     upper
## var1 0.3596919 0.8585628
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2007a_intslope)
```

```
##     var1
## 0.751598
```

```r
HPDinterval(cor_BV_RR_2007a_intslope)
```

```
##          lower     upper
## var1 0.4084901 0.8768752
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2008a_intslope)
```

```
##      var1
## 0.7326989
```

```r
HPDinterval(cor_BV_RR_2008a_intslope)
```

```
##          lower     upper
## var1 0.3929251 0.8640059
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2009a_intslope)
```

```
##      var1
## 0.7334756
```

```r
HPDinterval(cor_BV_RR_2009a_intslope)
```

```
##          lower     upper
## var1 0.3816495 0.8807742
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2010a_intslope)
```

```
##     var1
## 0.662941
```

```r
HPDinterval(cor_BV_RR_2010a_intslope)
```

```
##       lower     upper
## var1 0.3854 0.8618329
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2011a_intslope)
```

```
##      var1
## 0.7092884
```

```r
HPDinterval(cor_BV_RR_2011a_intslope)
```

```
##          lower     upper
## var1 0.3868073 0.8748384
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2012a_intslope)
```

```
##     var1
## 0.694108
```

```r
HPDinterval(cor_BV_RR_2012a_intslope)
```

```
##          lower     upper
## var1 0.3765925 0.8697939
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2013a_intslope)
```

```
##      var1
## 0.6853881
```

```r
HPDinterval(cor_BV_RR_2013a_intslope)
```

```
##          lower     upper
## var1 0.3642475 0.8720935
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2014a_intslope)
```

```
##      var1
## 0.6412917
```

```r
HPDinterval(cor_BV_RR_2014a_intslope)
```

```
##          lower    upper
## var1 0.3393826 0.847091
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2015a_intslope)
```

```
##      var1
## 0.6655729
```

```r
HPDinterval(cor_BV_RR_2015a_intslope)
```

```
##          lower     upper
## var1 0.3342386 0.8680274
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2016a_intslope)
```

```
##      var1
## 0.6227786
```

```r
HPDinterval(cor_BV_RR_2016a_intslope)
```

```
##          lower     upper
## var1 0.3529111 0.8491815
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2017a_intslope)
```

```
##      var1
## 0.7020102
```

```r
HPDinterval(cor_BV_RR_2017a_intslope)
```

```
##          lower     upper
## var1 0.3868558 0.8730766
## attr(,"Probability")
## [1] 0.95
```

Old period: No significant correlation among intercepts and slopes. New period: Significant positive correlation among intercepts and slopes.

```r
posterior.mode(cor_BV_RR_1987a_intfit)
```

Among-individual correlation between FFD and fitness:

```
##      var1
## 0.4157824
```

```r
HPDinterval(cor_BV_RR_1987a_intfit)
```

```
##           lower     upper
## var1 -0.1679743 0.8012159
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1988a_intfit)
```

```
##      var1
## 0.5060078
```

```r
HPDinterval(cor_BV_RR_1988a_intfit)
```

```
##           lower     upper
## var1 -0.1659891 0.8081149
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1989a_intfit)
```

```
##       var1
## -0.6213415
```

```r
HPDinterval(cor_BV_RR_1989a_intfit)
```

```
##          lower       upper
## var1 -0.880903 -0.08237789
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1990a_intfit)
```

```
##       var1
## -0.4771504
```

```r
HPDinterval(cor_BV_RR_1990a_intfit)
```

```
##           lower     upper
## var1 -0.7948577 0.2251857
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1991a_intfit)
```

```
##       var1
## -0.1810211
```

```r
HPDinterval(cor_BV_RR_1991a_intfit)
```

```
##           lower     upper
## var1 -0.5799095 0.5163885
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1992a_intfit)
```

```
##        var1
## -0.04348555
```

```r
HPDinterval(cor_BV_RR_1992a_intfit)
```

```
##           lower     upper
## var1 -0.6453459 0.5777166
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1993a_intfit)
```

```
##        var1
## -0.07802341
```

```r
HPDinterval(cor_BV_RR_1993a_intfit)
```

```
##           lower     upper
## var1 -0.7318247 0.4202965
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1994a_intfit)
```

```
##       var1
## -0.5600849
```

```r
HPDinterval(cor_BV_RR_1994a_intfit)
```

```
##           lower      upper
## var1 -0.9022327 0.07421195
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1995a_intfit)
```

```
##        var1
## -0.03380031
```

```r
HPDinterval(cor_BV_RR_1995a_intfit)
```

```
##           lower     upper
## var1 -0.7963906 0.6619564
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1996a_intfit)
```

```
##      var1
## 0.5322722
```

```r
HPDinterval(cor_BV_RR_1996a_intfit)
```

```
##           lower     upper
## var1 -0.2547201 0.9023993
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2006a_intfit)
```

```
##      var1
## -0.461533
```

```r
HPDinterval(cor_BV_RR_2006a_intfit)
```

```
##           lower      upper
## var1 -0.7413751 -0.1936854
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2007a_intfit)
```

```
##       var1
## -0.5488071
```

```r
HPDinterval(cor_BV_RR_2007a_intfit)
```

```
##           lower      upper
## var1 -0.7570265 -0.2770966
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2008a_intfit)
```

```
##       var1
## -0.1292851
```

```r
HPDinterval(cor_BV_RR_2008a_intfit)
```

```
##           lower     upper
## var1 -0.4386834 0.1110102
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2009a_intfit)
```

```
##       var1
## -0.5290712
```

```r
HPDinterval(cor_BV_RR_2009a_intfit)
```

```
##           lower      upper
## var1 -0.8068484 -0.1463703
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2010a_intfit)
```

```
##       var1
## -0.2466276
```

```r
HPDinterval(cor_BV_RR_2010a_intfit)
```

```
##           lower      upper
## var1 -0.5514517 0.08338056
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2011a_intfit)
```

```
##       var1
## -0.5353704
```

```r
HPDinterval(cor_BV_RR_2011a_intfit)
```

```
##           lower      upper
## var1 -0.7686083 -0.1193529
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2012a_intfit)
```

```
##       var1
## -0.5146205
```

```r
HPDinterval(cor_BV_RR_2012a_intfit)
```

```
##           lower      upper
## var1 -0.7867097 -0.2322993
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2013a_intfit)
```

```
##       var1
## -0.1333686
```

```r
HPDinterval(cor_BV_RR_2013a_intfit)
```

```
##           lower     upper
## var1 -0.6109703 0.3804461
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2014a_intfit)
```

```
##       var1
## -0.5814464
```

```r
HPDinterval(cor_BV_RR_2014a_intfit)
```

```
##           lower      upper
## var1 -0.7873738 -0.2250555
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2015a_intfit)
```

```
##       var1
## 0.05236346
```

```r
HPDinterval(cor_BV_RR_2015a_intfit)
```

```
##           lower     upper
## var1 -0.3511349 0.4550417
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2016a_intfit)
```

```
##       var1
## -0.4911559
```

```r
HPDinterval(cor_BV_RR_2016a_intfit)
```

```
##           lower      upper
## var1 -0.7684375 -0.2967115
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2017a_intfit)
```

```
##      var1
## -0.825888
```

```r
HPDinterval(cor_BV_RR_2017a_intfit)
```

```
##           lower       upper
## var1 -0.9618357 -0.08606709
## attr(,"Probability")
## [1] 0.95
```

Significant negative correlation among FFD and fitness in: 1989, 2006, 2007, 2009, 2011, 2012, 2014, 2016, 2017.

```r
posterior.mode(cor_BV_RR_1987a_slopefit)
```

Among-individual correlation between fitness and variation in slopes for FFD:

```
##        var1
## -0.01549488
```

```r
HPDinterval(cor_BV_RR_1987a_slopefit)
```

```
##           lower     upper
## var1 -0.6246535 0.4960935
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1988a_slopefit)
```

```
##       var1
## 0.02264473
```

```r
HPDinterval(cor_BV_RR_1988a_slopefit)
```

```
##           lower     upper
## var1 -0.6141592 0.5138305
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1989a_slopefit)
```

```
##       var1
## -0.2525233
```

```r
HPDinterval(cor_BV_RR_1989a_slopefit)
```

```
##           lower    upper
## var1 -0.7353075 0.291939
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1990a_slopefit)
```

```
##      var1
## -0.655067
```

```r
HPDinterval(cor_BV_RR_1990a_slopefit)
```

```
##           lower      upper
## var1 -0.8997189 -0.1611162
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1991a_slopefit)
```

```
##       var1
## 0.05648334
```

```r
HPDinterval(cor_BV_RR_1991a_slopefit)
```

```
##          lower    upper
## var1 -0.469841 0.596566
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1992a_slopefit)
```

```
##     var1
## 0.576571
```

```r
HPDinterval(cor_BV_RR_1992a_slopefit)
```

```
##           lower     upper
## var1 -0.2609525 0.8279511
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1993a_slopefit)
```

```
##      var1
## 0.2058132
```

```r
HPDinterval(cor_BV_RR_1993a_slopefit)
```

```
##          lower     upper
## var1 -0.488021 0.7165983
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1994a_slopefit)
```

```
##       var1
## -0.2734002
```

```r
HPDinterval(cor_BV_RR_1994a_slopefit)
```

```
##           lower     upper
## var1 -0.7443296 0.5221364
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1995a_slopefit)
```

```
##        var1
## -0.03842064
```

```r
HPDinterval(cor_BV_RR_1995a_slopefit)
```

```
##           lower     upper
## var1 -0.5707549 0.7555489
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_1996a_slopefit)
```

```
##       var1
## 0.04772102
```

```r
HPDinterval(cor_BV_RR_1996a_slopefit)
```

```
##           lower     upper
## var1 -0.6396896 0.6846672
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2006a_slopefit)
```

```
##       var1
## -0.1664045
```

```r
HPDinterval(cor_BV_RR_2006a_slopefit)
```

```
##           lower     upper
## var1 -0.5291767 0.2131438
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2007a_slopefit)
```

```
##       var1
## -0.5742125
```

```r
HPDinterval(cor_BV_RR_2007a_slopefit)
```

```
##           lower      upper
## var1 -0.7415447 -0.1954424
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2008a_slopefit)
```

```
##       var1
## -0.3800705
```

```r
HPDinterval(cor_BV_RR_2008a_slopefit)
```

```
##           lower       upper
## var1 -0.6400981 -0.05228276
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2009a_slopefit)
```

```
##       var1
## -0.4690445
```

```r
HPDinterval(cor_BV_RR_2009a_slopefit)
```

```
##           lower       upper
## var1 -0.7772853 -0.03102824
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2010a_slopefit)
```

```
##        var1
## -0.09721649
```

```r
HPDinterval(cor_BV_RR_2010a_slopefit)
```

```
##           lower     upper
## var1 -0.4891571 0.2174477
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2011a_slopefit)
```

```
##       var1
## -0.5228365
```

```r
HPDinterval(cor_BV_RR_2011a_slopefit)
```

```
##           lower       upper
## var1 -0.7952939 -0.08264562
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2012a_slopefit)
```

```
##       var1
## -0.1511505
```

```r
HPDinterval(cor_BV_RR_2012a_slopefit)
```

```
##           lower     upper
## var1 -0.4863352 0.2483004
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2013a_slopefit)
```

```
##      var1
## 0.1000533
```

```r
HPDinterval(cor_BV_RR_2013a_slopefit)
```

```
##           lower     upper
## var1 -0.4456479 0.5799181
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2014a_slopefit)
```

```
##       var1
## -0.1918313
```

```r
HPDinterval(cor_BV_RR_2014a_slopefit)
```

```
##           lower     upper
## var1 -0.5301463 0.2809885
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2015a_slopefit)
```

```
##      var1
## 0.4829326
```

```r
HPDinterval(cor_BV_RR_2015a_slopefit)
```

```
##            lower     upper
## var1 -0.03315891 0.7014147
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2016a_slopefit)
```

```
##      var1
## -0.108642
```

```r
HPDinterval(cor_BV_RR_2016a_slopefit)
```

```
##           lower     upper
## var1 -0.4973761 0.1867981
## attr(,"Probability")
## [1] 0.95
```

```r
posterior.mode(cor_BV_RR_2017a_slopefit)
```

```
##       var1
## -0.6384911
```

```r
HPDinterval(cor_BV_RR_2017a_slopefit)
```

```
##           lower     upper
## var1 -0.8629713 0.0733726
## attr(,"Probability")
## [1] 0.95
```

Significant negative correlation among FFD and variation in slopes for FFD in: 1990, 2007, 2008, 2009, 2011.

```r
summary(modelBV_RR_1987a)$solutions
```

Fixed effects

```
##                                  post.mean   l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                     59.6448353 54.5697785 64.3913623 2145.221 0.0005
## variablefitness                  0.1227164 -0.8645798  0.9831157 2000.000 0.7550
## at.level(variable, "FFD"):temp -2.0224455 -6.0417310  1.7577306 1708.336 0.2710
```

```r
summary(modelBV_RR_1988a)$solutions
```

```
##                                  post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                     59.5711468 54.751468 64.7893447 2000.000 0.0005
## variablefitness                 -0.5661771 -1.502892  0.3011084 2000.000 0.2140
## at.level(variable, "FFD"):temp -2.0790485 -6.154478  2.0094366 1755.867 0.2760
```

```r
summary(modelBV_RR_1989a)$solutions
```

```
##                                  post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                     59.7112795 54.662102 64.3662266 2000.000 0.0005
## variablefitness                 -0.0540044 -1.033793  0.9340941 2040.495 0.9610
## at.level(variable, "FFD"):temp -2.0021080 -5.765733  2.0828187 2000.000 0.2890
```

```r
summary(modelBV_RR_1990a)$solutions
```

```
##                                 post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                     59.765226 55.053708 65.2979377 2000.000 0.0005
## variablefitness                 -2.223500 -3.767580 -0.8041527 2000.000 0.0005
## at.level(variable, "FFD"):temp -2.042081 -6.195027  1.9716244 1979.187 0.3130
```

```r
summary(modelBV_RR_1991a)$solutions
```

```
##                                  post.mean   l-95% CI  u-95% CI eff.samp  pMCMC
## variableFFD                     59.7203919 54.3933549 64.137737     2000 0.0005
## variablefitness                  0.6515546 -0.0669377  1.272210     2000 0.0760
## at.level(variable, "FFD"):temp -2.0286656 -6.1485973  1.792945     2000 0.2700
```

```r
summary(modelBV_RR_1992a)$solutions
```

```
##                                 post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## variableFFD                     59.697902 54.714461 65.007540 1934.931 0.0005
## variablefitness                 -2.283734 -3.792540 -1.024416 2000.000 0.0005
## at.level(variable, "FFD"):temp -2.040294 -6.155864  1.790436 2000.000 0.2890
```

```r
summary(modelBV_RR_1993a)$solutions
```

```
##                                  post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                      59.656417 54.518815 65.0164425 2203.309 0.0005
## variablefitness                  -1.978477 -3.384998 -0.5043028 2286.341 0.0005
## at.level(variable, "FFD"):temp -2.042207 -6.158129  1.9146966 2144.270 0.2870
```

```r
summary(modelBV_RR_1994a)$solutions
```

```
##                                  post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## variableFFD                      59.624961 54.839867 65.386976 2000.000 0.0005
## variablefitness                  -3.770788 -6.218526 -1.709156 2000.000 0.0005
## at.level(variable, "FFD"):temp -2.066376 -6.494987  2.034188 2136.482 0.2970
```

```r
summary(modelBV_RR_1995a)$solutions
```

```
##                                  post.mean   l-95% CI  u-95% CI eff.samp  pMCMC
## variableFFD                      59.588457  54.436311 64.637996 2000.000 0.0005
## variablefitness                  -6.950456 -12.236869 -2.982198 2193.023 0.0005
## at.level(variable, "FFD"):temp -2.130220  -5.855962  1.932219 2000.000 0.2630
```

```r
summary(modelBV_RR_1996a)$solutions
```

```
##                                  post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## variableFFD                      59.608513 54.568614 64.454564 1972.047 0.0005
## variablefitness                  -4.452778 -7.161676 -1.907528 2000.000 0.0005
## at.level(variable, "FFD"):temp -2.082465 -6.401969  1.888749 1807.617 0.2830
```

```r
summary(modelBV_RR_2006a)$solutions
```

```
##                                  post.mean  l-95% CI   u-95% CI eff.samp pMCMC
## variableFFD                      55.474392 53.731284 57.2648606     2000 5e-04
## variablefitness                  -1.755720 -2.989685 -0.6266602     2000 1e-03
## at.level(variable, "FFD"):temp -1.723939 -2.958892 -0.3907249     2000 8e-03
```

```r
summary(modelBV_RR_2007a)$solutions
```

```
##                                   post.mean   l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                      55.4327603 53.7946309 57.2701373 2000.000 0.0005
## variablefitness                   0.1973966 -0.3799148  0.7475721 2000.000 0.4900
## at.level(variable, "FFD"):temp -1.6729515 -2.8423428 -0.3843603 2328.332 0.0140
```

```r
summary(modelBV_RR_2008a)$solutions
```

```
##                                   post.mean   l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                      55.4424175 53.5589531 57.1920450 2000.000 0.0005
## variablefitness                   0.8386921  0.2067913  1.4469589 2162.489 0.0200
## at.level(variable, "FFD"):temp -1.6561856 -2.9040655 -0.4937203 2000.000 0.0160
```

```r
summary(modelBV_RR_2009a)$solutions
```

```
##                                   post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.427784 53.643117 57.1779391 2000.000 0.0005
## variablefitness                   -2.400046 -3.411977 -1.4294945 2036.341 0.0005
## at.level(variable, "FFD"):temp -1.659183 -3.025379 -0.4938329 2000.000 0.0170
```

```r
summary(modelBV_RR_2010a)$solutions
```

```
##                                    post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.4443770 53.694189 57.3661891 2173.284 0.0005
## variablefitness                   -0.9531035 -1.611459 -0.2957857 2060.871 0.0010
## at.level(variable, "FFD"):temp -1.6719604 -2.847898 -0.3328963 2000.000 0.0160
```

```r
summary(modelBV_RR_2011a)$solutions
```

```
##                                   post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.452748 53.851213 57.2402043 2000.000 0.0005
## variablefitness                   -1.988282 -2.870297 -1.2193500 2000.000 0.0005
## at.level(variable, "FFD"):temp -1.660783 -2.856271 -0.4552464 1797.787 0.0130
```

```r
summary(modelBV_RR_2012a)$solutions
```

```
##                                   post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.452671 53.775794 57.2293844 2000.000 0.0005
## variablefitness                   -3.025701 -4.738324 -1.5531132 1497.546 0.0005
## at.level(variable, "FFD"):temp -1.678298 -2.895469 -0.3622153 2288.492 0.0160
```

```r
summary(modelBV_RR_2013a)$solutions
```

```
##                                   post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.427107 53.706798 57.2389522     2000 0.0005
## variablefitness                   -4.705340 -7.241894 -2.5447426     2000 0.0005
## at.level(variable, "FFD"):temp -1.675413 -2.916656 -0.4022641     2000 0.0130
```

```r
summary(modelBV_RR_2014a)$solutions
```

```
##                                   post.mean  l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.451947 53.406437 57.1147306     2000 0.0005
## variablefitness                   -3.084868 -4.707263 -1.6942112     2000 0.0005
## at.level(variable, "FFD"):temp -1.671774 -2.850009 -0.4130259     2000 0.0120
```

```r
summary(modelBV_RR_2015a)$solutions
```

```
##                                   post.mean   l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                       55.398966  53.682364 57.3583201 1819.638 0.0005
## variablefitness                   -6.844942 -10.901900 -3.8960190 2000.000 0.0005
## at.level(variable, "FFD"):temp -1.692280  -3.001696 -0.4563431 1620.584 0.0140
```

```r
summary(modelBV_RR_2016a)$solutions
```

```
##                                   post.mean   l-95% CI   u-95% CI eff.samp  pMCMC
## variableFFD                      55.4462026 53.8034152 57.2792895     2000 0.0005
## variablefitness                   0.1429394 -0.5358374  0.7910238     2000 0.6460
## at.level(variable, "FFD"):temp -1.6856854 -2.9122095 -0.4108379     2000 0.0150
```

```r
summary(modelBV_RR_2017a)$solutions
```

```
##                                   post.mean   l-95% CI  u-95% CI  eff.samp  pMCMC
## variableFFD                       55.419801  53.652598 57.182886 1779.2192 0.0005
## variablefitness                  -12.313037 -24.492693 -4.019905  329.2653 0.0005
## at.level(variable, "FFD"):temp  -1.673102  -2.840028 -0.456684 2000.0000 0.0140
```

Significant fixed effect of temperature in all years from the new period, but in none from the old period.

With shoot volume

Worth doing?

```r
pbPost(title="Done!")
```

TO DO

- Include interaction between year and intercept/slope in the random part of the MCMCglmm models: to see if selection varies among years (instead of doing yearly models)
- Include interaction between size and intercept/slope in the random part of the MCMCglmm models: to see if selection varies with size (in the same model, without using the BLUPs as above)