

Lathyrus ms2: Selection on reaction norms - multivariate modeling for phenotypic selection on plasticity 3 (Arnold et al. 2019 Phil. Trans. R. Soc. B)

Contents

Using ids with 4 years of data	2
Total fitness	2
Temperature: mean April	2
Without number of flowers	2
With number of flowers	12
Temperature: mean May	14
Temperature: min April	16
Temperature: mean April-May	18
Mean fitness per flowering event	20
Temperature: mean April	20
Without number of flowers	20
With number of flowers	22
Temperature: mean May	24
Temperature: min April	26
Temperature: mean April-May	28
Summary 4 years	30
Using ids with 5 years of data	31
Total fitness	31
Temperature: mean April	31
Without number of flowers	31
With number of flowers	33
Temperature: mean May	35
Mean fitness per flowering event	37
Temperature: mean April	37
Temperature: mean May	39
Summary 5 years	41
Only model with selection on RN slope: modelBV_RR_pois_5	42

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 either total fitness or mean fitness per flowering event. Trying different alternatives for temperature (mean April, mean May, min April, mean April-May). Trying also to include number of flowers.

Using ids with 4 years of data

Total fitness

Temperature: mean April

Without number of flowers

Stack data:

The bivariate MCMCglmm models with covariances between fitness and FFD intercepts and slopes are run using a stacked data set. We therefore need to first stack our data and add in index columns of 'traits', 'variable', and 'family', which are all required for MCMCglmm to fit a multivariate model of traits with different distributions. The values of the two variables need to be stacked into a single column and identified as either 'fitness' or 'FFD' with an index column. Likewise, the 'family' index column tells MCMCglmm what distribution family that response variable follows (here, Poisson or Gaussian). For this data-set, we put the fitness data first, then the repeated-measures FFD data.

```
# Create a single data-set "data.stack_pois", with single column at start to index observations
data.stack_pois <- c()
data.stack_pois$Obs <- 1:(243 + 1455)
data.stack_pois$id <- c(data_4yrs_total$id, data_4yrs$id)

# Add first_yr to total data +
# Year column is only relevant for FFD, but is set to first_yr for fitness values

data_4yrs_total_wfirstyr <- data_4yrs_total %>%
  right_join(data_4yrs[c(3,10)] %>%
    group_by(id) %>%
    summarise(first_yr = mean(first_yr)), by = "id")

data.stack_pois$year <- c(data_4yrs_total_wfirstyr$first_yr,
  data_4yrs$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois$temp <- c(rep(0, 243), data_4yrs$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois$variable <- data.stack_pois$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois <- data.frame(data.stack_pois)

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

```
##   Obs id year temp fitness.FFD.stack  traits variable family
## 1    1 1 2006    0                14 fitness  fitness poisson
## 2    2 2 2007    0                4  fitness  fitness poisson
```

```
## 3 3 3 2007 0 2 fitness fitness poisson
## 4 4 4 2007 0 6 fitness fitness poisson
## 5 5 5 2007 0 4 fitness fitness poisson
## 6 6 6 2007 0 2 fitness fitness poisson

# Scaling factor for MCMCglmm iterations
sc <- 100#0 # Increase this parameter for longer runs

priorBiv_RR_pois <- 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_RR_pois <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

kable(summary(modelBV_RR_pois)$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	58.129	55.994	60.098	1000	0.001
variablefitness	0.974	0.844	1.088	1277	0.001
at.level(variable, "FFD"):temp	-2.417	-3.922	-0.799	1000	0.004

```
kable(summary(modelBV_RR_pois)$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	26.45	12.739	45.378	1000

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

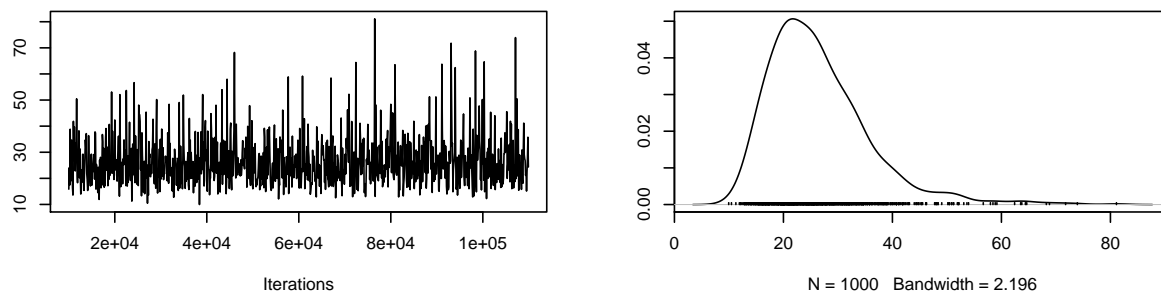
Table 3: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, "FFD").id:at.level(variable, "FFD").id	3.092	1.913	4.200	1102
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	0.855	0.367	1.323	1000
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.650	-0.953	-0.342	1000
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	0.855	0.367	1.323	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	0.717	0.369	1.112	1000
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.128	-0.300	0.050	1197
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.650	-0.953	-0.342	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.128	-0.300	0.050	1197
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.524	0.374	0.681	1066
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	18.599	17.086	20.257	1262

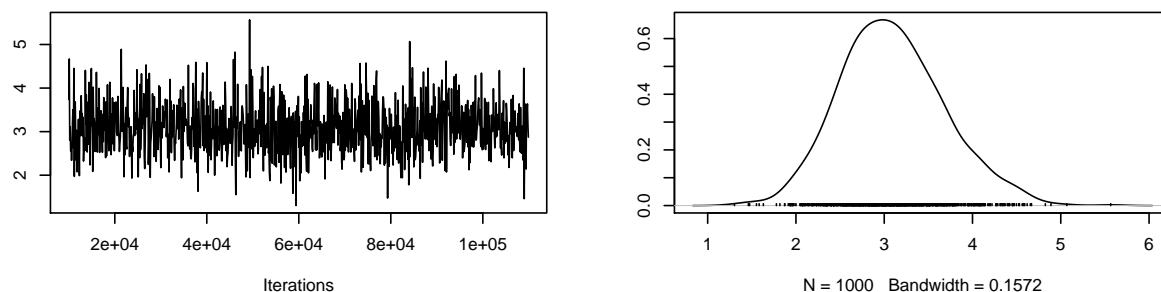
Meaning of each row: "Among-id var in intercept for FFD", "Among-id covar between intercept and slope for FFD", "Among-individual covar between fitness and var in intercepts for FFD", "Among-id covar between intercept and slope for FFD", "Among-id var in slope for FFD", "Among-id covar between slope for FFD and intercept for fitness", "Among-individual covar between fitness and var in intercepts for FFD", "Among-id covar between slope for FFD and intercept for fitness", "Among-id var in intercept for fitness", "Within-id var in FFD between years".

```
plot(modelBV_RR_pois$VCV[,1:4])
```

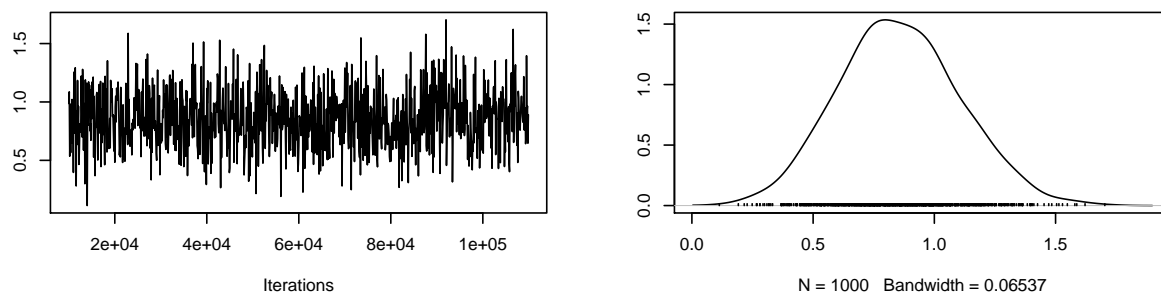
Trace of at.level(variable, "FFD"):at.level(variable, "FFD").ye Density of at.level(variable, "FFD"):at.level(variable, "FFD").y



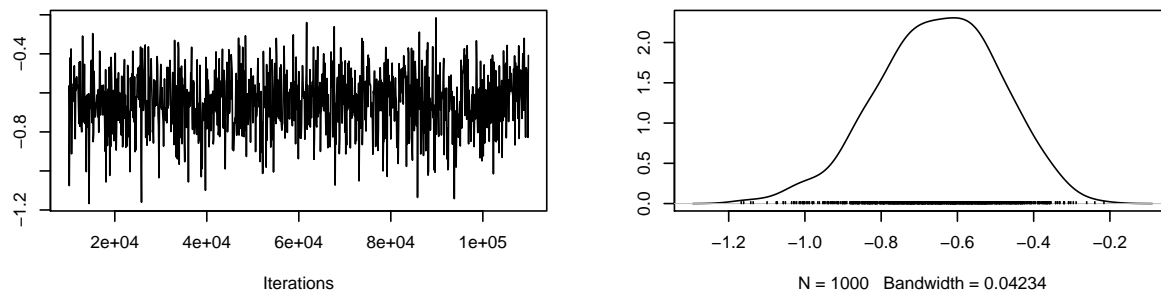
Trace of at.level(variable, "FFD").id:at.level(variable, "FFD") Density of at.level(variable, "FFD").id:at.level(variable, "FFD")



Trace of at.level(variable, "FFD"):temp.id:at.level(variable, "FFD") Density of at.level(variable, "FFD"):temp.id:at.level(variable, "FFD")

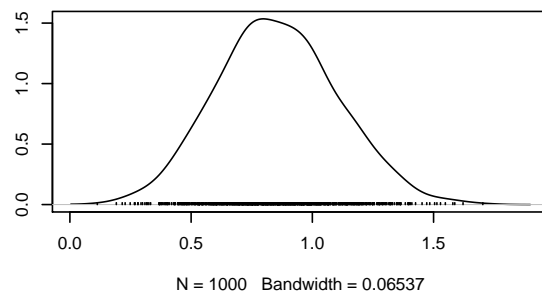
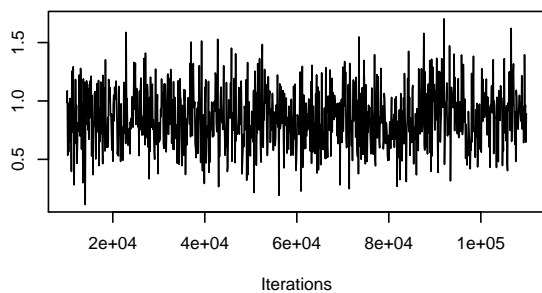


Trace of at.level(variable, "fitness").id:at.level(variable, "FFD") Density of at.level(variable, "fitness").id:at.level(variable, "FFD")

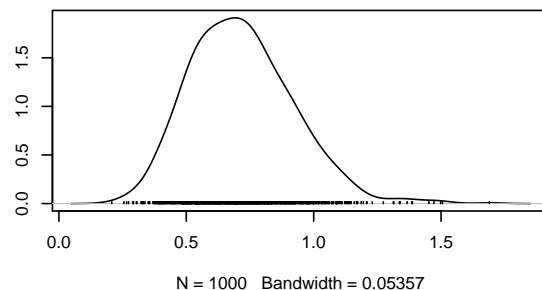
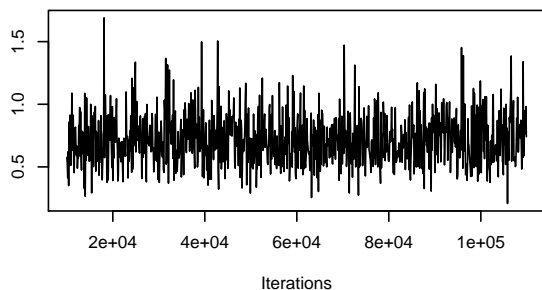


```
plot(modelBV_RR_pois$VCV[,5:8])
```

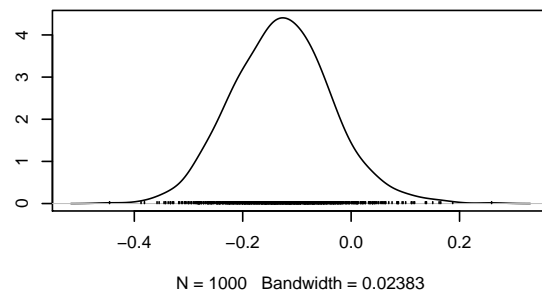
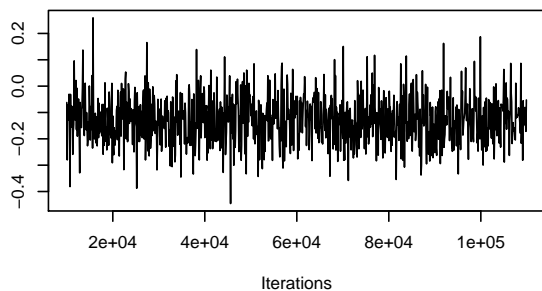
race of at.level(variable, "FFD").id:at.level(variable, "FFD"):ten
sity of at.level(variable, "FFD").id:at.level(variable, "FFD"):ten



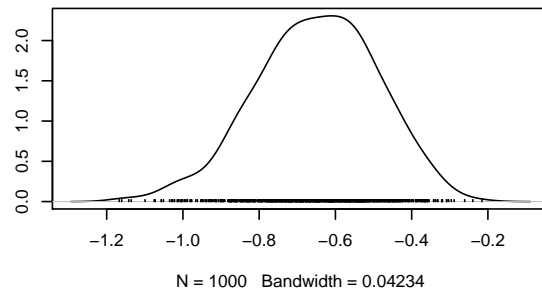
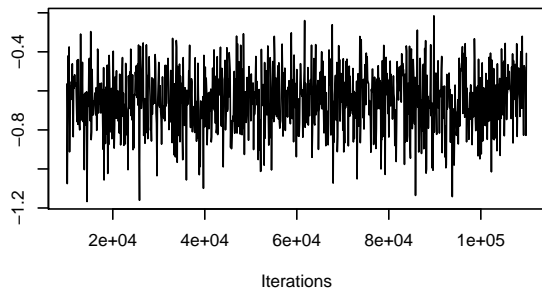
e of at.level(variable, "FFD"):temp.id:at.level(variable, "FFD")ity of at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"



ice of at.level(variable, "fitness").id:at.level(variable, "FFD"):t:city of at.level(variable, "fitness").id:at.level(variable, "FFD"):

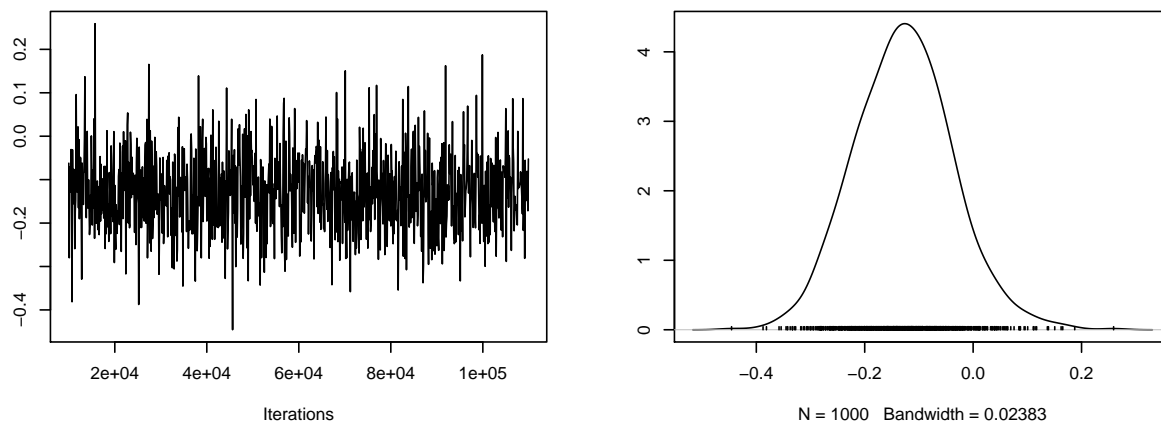


Trace of at.level(variable, "FFD").id:at.level(variable, "fitness")density of at.level(variable, "FFD").id:at.level(variable, "fitness")

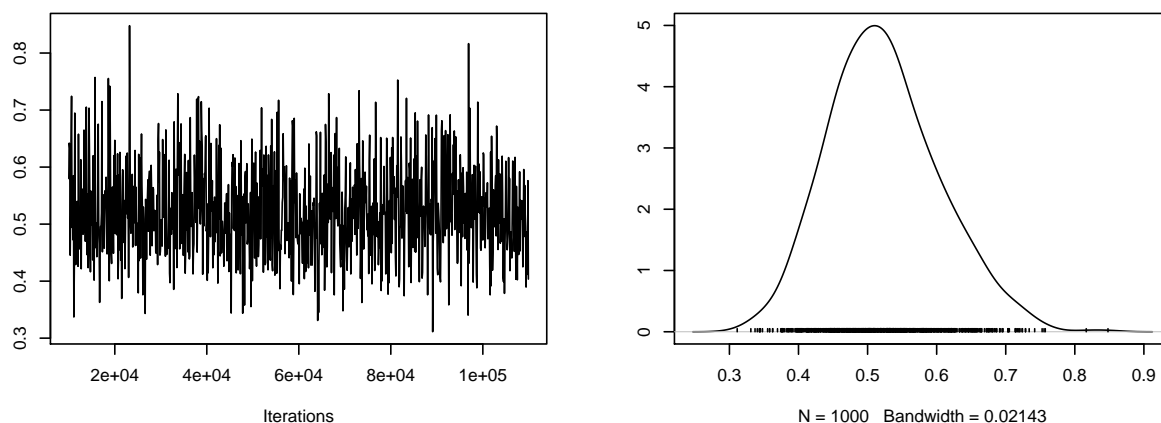


```
plot(modelBV_RR_pois$VCV[,9:11])
```

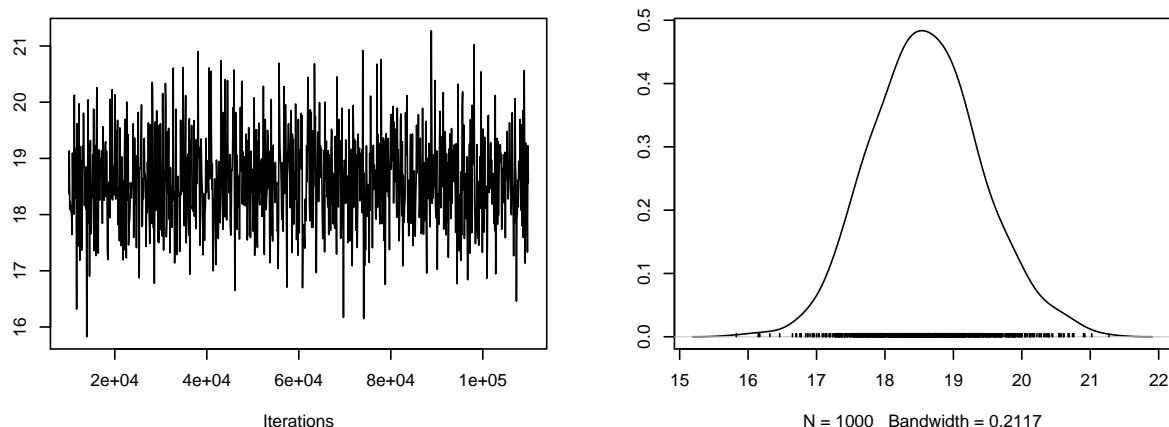
ice of at.level(variable, "FFD"):temp.id:at.level(variable, "fitness) of at.level(variable, "FFD"):temp.id:at.level(variable, "fitn



race of at.level(variable, "fitness").id:at.level(variable, "fitness").density of at.level(variable, "fitness").id:at.level(variable, "fitness").density



Trace of at.level(variable, "FFD"):at.level(variable, "FFD").O Density of at.level(variable, "FFD"):at.level(variable, "FFD").(



Check for autocorrelation between successive stored iterations (suggested to be less than 0.1):

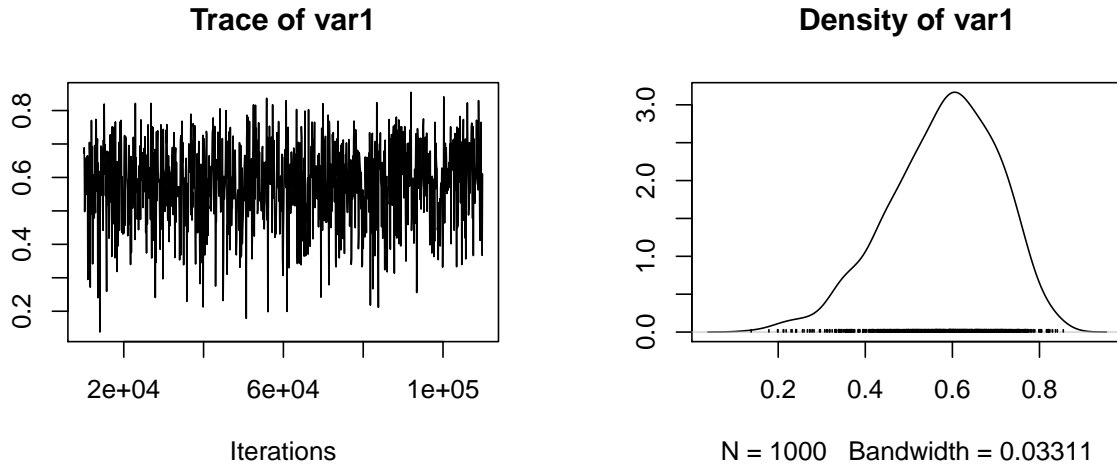
```
kable(diag(autocorr(modelBV_RR_pois$VCV)[2, , ]),caption="Autocorrelation")
```

Table 4: Autocorrelation

	x
at.level(variable, "FFD"):at.level(variable, "FFD").year	-0.0091692
at.level(variable, "FFD").id:at.level(variable, "FFD").id	-0.0489505
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	-0.0248751
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.0258161
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	-0.0248751
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	-0.0187903
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.0331344
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.0258161
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.0331344
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.0337469
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	0.0067911

Ensure that the among-individual correlation between intercepts and slopes for FFD is (approximately) the same as we estimated in our earlier univariate random regression model.

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



```
posterior.mode(cor_BV_RR_pois_intslope)
```

```
##      var1
## 0.594561
```

```
HPDinterval(cor_BV_RR_pois_intslope)
```

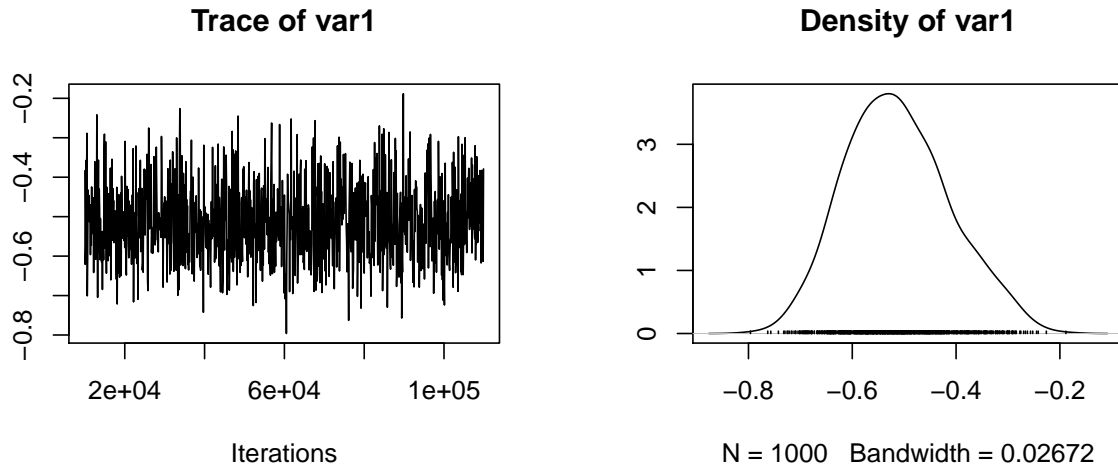
```
##      lower      upper
## var1 0.3298192 0.7825926
## attr("Probability")
## [1] 0.95
```

We find a strong positive correlation between among-individual variance in intercepts and slopes, at the

intercept ($x = 0$). Although it is a bit lower than in our earlier univariate random regression model (OK?).

Determining the among-individual correlation between FFD and fitness:

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



```
posterior.mode(cor_BV_RR_pois_intfit)
```

```
##      var1
## -0.5706904
```

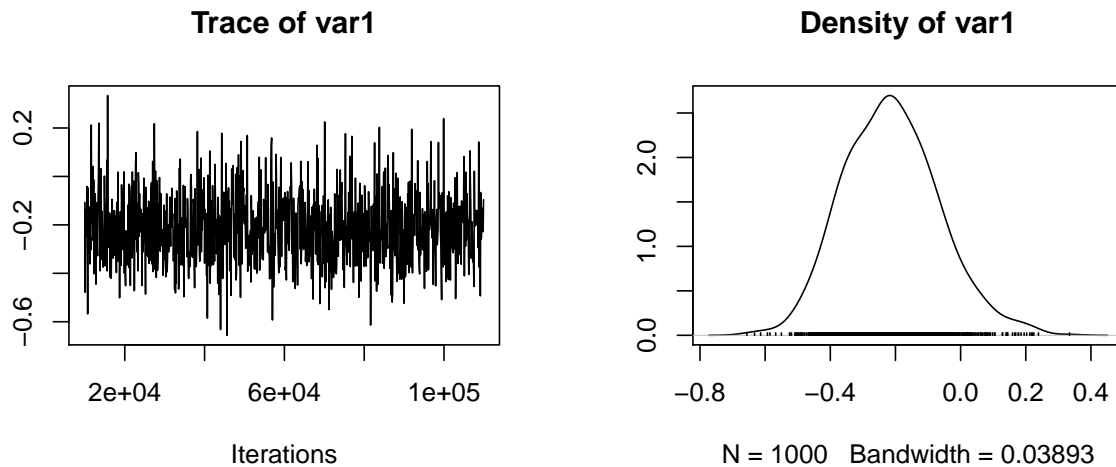
```
HPDinterval(cor_BV_RR_pois_intfit)
```

```
##      lower      upper
## var1 -0.6970216 -0.3091069
## attr("Probability")
## [1] 0.95
```

Negative correlation: Fitness increases when FFD decreases (i.e. is earlier).

Determining the among-individual correlation between fitness and variation in slopes for FFD:

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



```
posterior.mode(cor_BV_RR_pois_slopefit)
```

```
##      var1
## -0.2251261
```

```
HPDinterval(cor_BV_RR_pois_slopefit)
```

```
##      lower      upper
## var1 -0.5012255 0.0667102
## attr("Probability")
## [1] 0.95
```

Negative correlation: Fitness increases when the slope for FFD decreases (i.e. is more negative, and therefore plasticity increases). Fitness is higher in more plastic individuals. However, this correlation is not significant because the CIs encompass zero!!!

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.

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

```
##      [,1]      [,2]      [,3]
## [1,] 2.8949557 0.7508321 -0.6104302
```

```
## [2,] 0.7508321 0.6280586 -0.1135805
## [3,] -0.6104302 -0.1135805 0.5308708
```

```
# Extract selection *differentials* (i.e. covariances) for intercept and slope:
# and calculate posterior mode and credible intervals for each
```

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

```
## [1] -0.6104302 -0.1135805
```

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

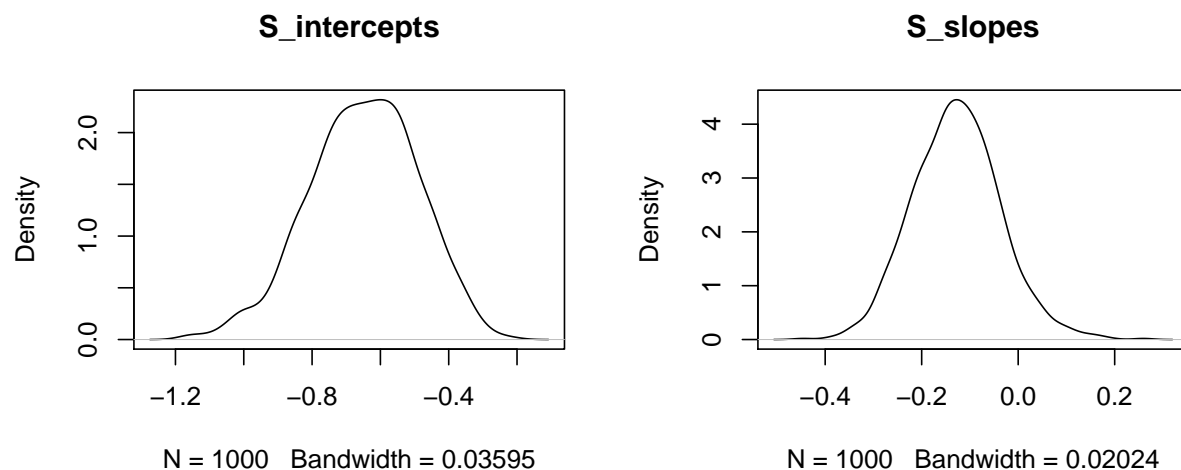
```
## S_intercepts S_slopes
## -0.6104302 -0.1135805
```

```
HPDinterval(mcmc(S.modelBV_RR_pois))
```

```
## lower upper
## S_intercepts -0.9529606 -0.34190093
## S_slopes -0.3003430 0.05017838
## attr("Probability")
## [1] 0.95
```

```
# Plot posterior distribution of selection differentials
```

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



```
# Estimate selection gradients for intercept and slope (beta = S / P)
```

```
# on each sample of posterior and extract their mode
```

```
n <- length(modelBV_RR_pois$VCV[,2]) # sample size
beta_post_RR_pois <- 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_RR_pois[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_RR_pois[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_RR_pois) <- c("beta_intercepts", "beta_slopes")
posterior.mode(mcmc(beta_post_RR_pois))

```

```

## beta_intercepts    beta_slopes
##      -0.2569895      0.1445244

```

```

HPDinterval(mcmc(beta_post_RR_pois))

```

```

##                lower      upper
## beta_intercepts -0.3733008 -0.1022348
## beta_slopes     -0.1892722  0.4764652
## attr("Probability")
## [1] 0.95

```

```

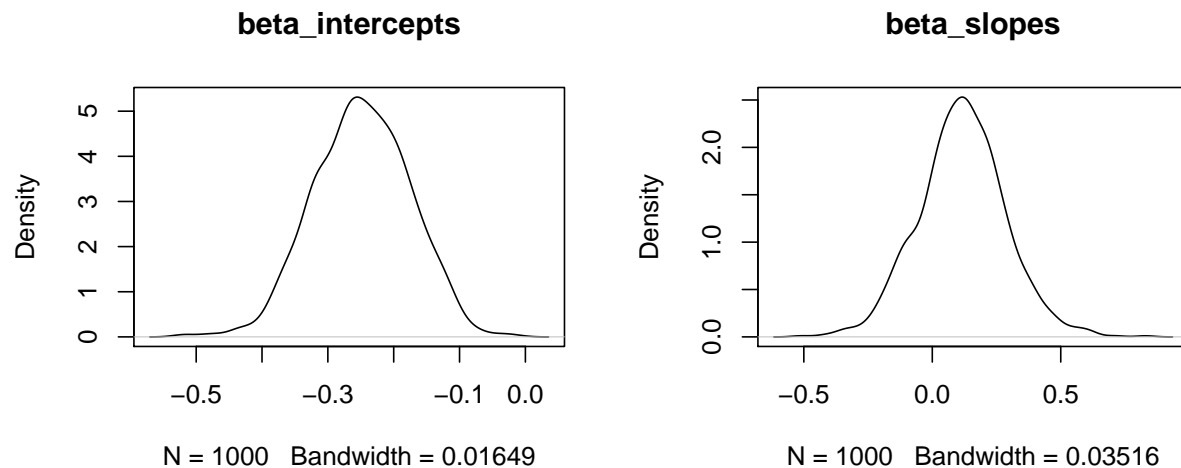
# Plot posterior distribution of selection gradients

```

```

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

```

With number of flowers

Stack data:

```

# Create a single data-set "data.stack_pois_nfl",
# with single column at start to index observations
data.stack_pois_nfl <- c()
data.stack_pois_nfl$Obs <- 1:(243 + 1455)
data.stack_pois_nfl$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois_nfl$year <- c(data_4yrs_total_wfirstyr$first_yr,
                             data_4yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_nfl$temp <- c(rep(0, 243), data_4yrs$cmean_4)

# Number of flowers column is only relevant for FFD, but is set to 0 for fitness values
data_4yrs$cn_fl <- scale(data_4yrs$n_fl, center=T, scale=F) # Standardize number of flowers
data.stack_pois_nfl$n_fl <- c(rep(0, 243), data_4yrs$cn_fl)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_nfl$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois_nfl$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois_nfl$variable <- data.stack_pois_nfl$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois_nfl$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois_nfl <- data.frame(data.stack_pois_nfl)

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

```

```

##   Obs id year temp n_fl fitness.FFD.stack  traits variable family
## 1    1  1 2006    0    0                14 fitness  fitness poisson
## 2    2  2 2007    0    0                 4 fitness  fitness poisson
## 3    3  3 2007    0    0                 2 fitness  fitness poisson
## 4    4  4 2007    0    0                 6 fitness  fitness poisson
## 5    5  5 2007    0    0                 4 fitness  fitness poisson
## 6    6  6 2007    0    0                 2 fitness  fitness poisson

```

```

modelBV_RR_pois_nfl <- 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, "FFD"):n_fl, # single fixed effect of n_fl
                               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 individuals in fitness
                               # (which is residual variance)
                               us(at.level(variable, "FFD")):Obs,

```

```

# ~ residual variance within individuals between years
# (labelled by 'Obs')
data = data.stack_pois_nfl,
prior = priorBiv_RR_pois,
family = NULL, # specified already in the data-set
nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

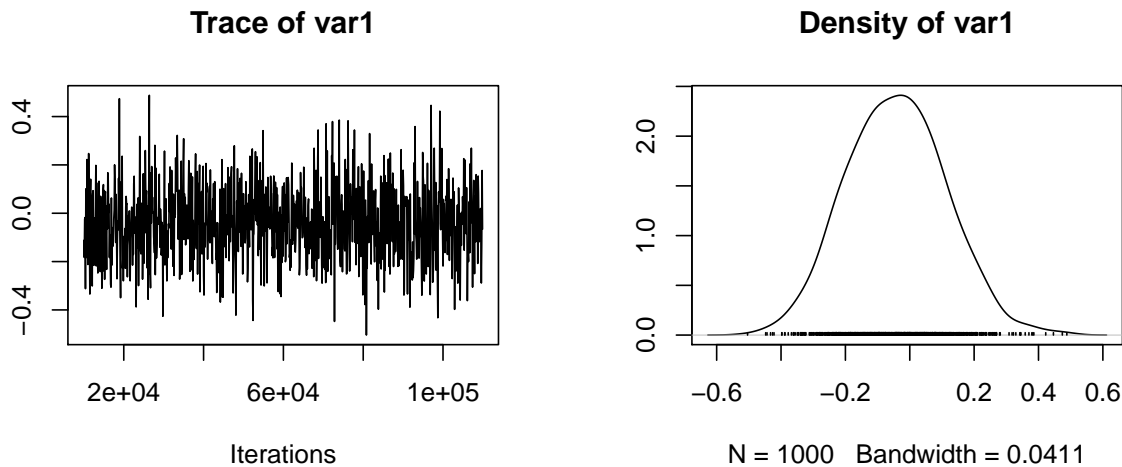
```

Determining the among-individual correlation between fitness and variation in slopes for FFD (we only focus in this result in this and the following models, to see if it reaches significance with any of the combinations):

```

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

```



```
posterior.mode(cor_BV_RR_pois_nfl_slopefit)
```

```
##          var1
## -0.04070236
```

```
HPDinterval(cor_BV_RR_pois_nfl_slopefit)
```

```
##          lower      upper
## var1 -0.3496906 0.2384892
## attr(,"Probability")
## [1] 0.95
```

Temperature: mean May

Mean-centred mean May temperature:

```

data_4yrs<-data_4yrs%>%
  mutate(cmean_5=scale(mean_5,center=T,scale=F))

```

Stack data:

```

# Create a single data-set "data.stack_pois_may", with single column at start to index observations
data.stack_pois_may <- c()
data.stack_pois_may$Obs <- 1:(243 + 1455)
data.stack_pois_may$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois_may$year <- c(data_4yrs_total_wfirstyr$first_yr,
                             data_4yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_may$temp <- c(rep(0, 243), data_4yrs$cmean_5)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_may$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois_may$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois_may$variable <- data.stack_pois_may$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois_may$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois_may <- data.frame(data.stack_pois_may)

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

```

```

##   Obs id year temp fitness.FFD.stack traits variable family
## 1   1  1 2006    0                14 fitness fitness poisson
## 2   2  2 2007    0                 4 fitness fitness poisson
## 3   3  3 2007    0                 2 fitness fitness poisson
## 4   4  4 2007    0                 6 fitness fitness poisson
## 5   5  5 2007    0                 4 fitness fitness poisson
## 6   6  6 2007    0                 2 fitness fitness poisson

```

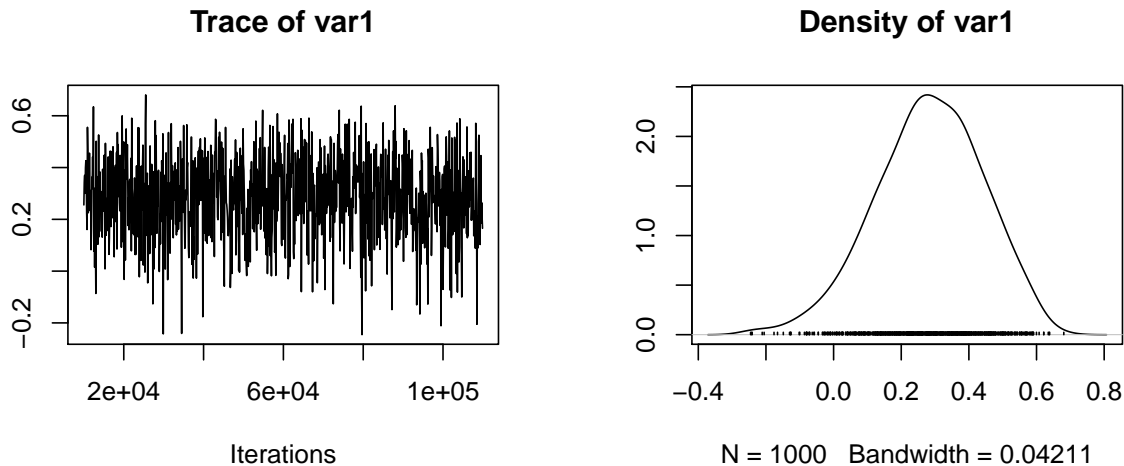
```

modelBV_RR_pois_may <- 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 individuals in fitness
                               # (which is residual variance)
                               us(at.level(variable, "FFD")):Obs,
                               # ^ residual variance within individuals between years
                               # (labelled by 'Obs')
                               data = data.stack_pois_may,
                               prior = priorBiv_RR_pois,
                               family = NULL, # specified already in the data-set
                               nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

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



```
posterior.mode(cor_BV_RR_pois_may_slopefit)
```

```
##      var1
## 0.279725
```

```
HPDinterval(cor_BV_RR_pois_may_slopefit)
```

```
##              lower      upper
## var1 -0.009500759 0.5907167
## attr("Probability")
## [1] 0.95
```

Temperature: min April

Mean-centred min April temperature:

```
data_4yrs <- data_4yrs %>%
  mutate(cmin_4 = scale(min_4, center=T, scale=F))
```

Stack data:

```
# Create a single data-set "data.stack_pois_min", with single column at start to index observations
data.stack_pois_min <- c()
data.stack_pois_min$Obs <- 1:(243 + 1455)
data.stack_pois_min$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois_min$year <- c(data_4yrs_total$first_yr,
  data_4yrs$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_min$temp <- c(rep(0, 243), data_4yrs$cmin_4)
```



```

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_min$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois_min$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois_min$variable <- data.stack_pois_min$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois_min$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois_min <- data.frame(data.stack_pois_min)

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

```

```

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0                14 fitness  fitness poisson
## 2    2  2 2007    0                 4 fitness  fitness poisson
## 3    3  3 2007    0                 2 fitness  fitness poisson
## 4    4  4 2007    0                 6 fitness  fitness poisson
## 5    5  5 2007    0                 4 fitness  fitness poisson
## 6    6  6 2007    0                 2 fitness  fitness poisson

```

```

modelBV_RR_pois_min <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois_min,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

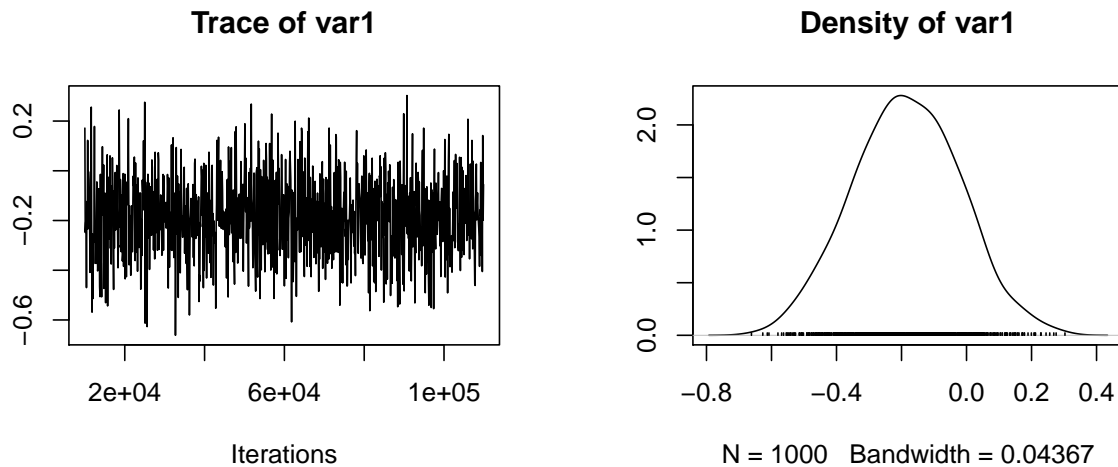
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

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

```



```
posterior.mode(cor_BV_RR_pois_min_slopefit)
```

```
##      var1
## -0.1997217
```

```
HPDinterval(cor_BV_RR_pois_min_slopefit)
```

```
##           lower      upper
## var1 -0.5142569 0.1259144
## attr("Probability")
## [1] 0.95
```

Temperature: mean April-May

Add April-May temperatures:

```
mean_weather_aprilmay$year<-as.integer(as.character(mean_weather_aprilmay$year))
data_4yrs<-data_4yrs%>%
  right_join(unique(mean_weather_aprilmay),by="year")%>%
  mutate(cmean_45=scale(mean_45,center=T,scale=F))
```

Stack data:

```
# Create a single data-set "data.stack_pois_aprilmay",
# with single column at start to index observations
data.stack_pois_aprilmay <- c()
data.stack_pois_aprilmay$Obs <- 1:(243 + 1455)
data.stack_pois_aprilmay$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois_aprilmay$year <- c(data_4yrs_total$firstyr$first_yr,
  data_4yrs$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_aprilmay$temp <- c(rep(0, 243), data_4yrs$cmean_45)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_aprilmay$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness), data_4yrs$FFD)
```

```

# Create 3 index columns needed for MCMCglmm
data.stack_pois_aprilmay$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois_aprilmay$variable <- data.stack_pois_aprilmay$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois_aprilmay$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois_aprilmay <- data.frame(data.stack_pois_aprilmay)

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

```

```

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0              14 fitness  fitness poisson
## 2    2  2 2007    0              4 fitness  fitness poisson
## 3    3  3 2007    0              2 fitness  fitness poisson
## 4    4  4 2007    0              6 fitness  fitness poisson
## 5    5  5 2007    0              4 fitness  fitness poisson
## 6    6  6 2007    0              2 fitness  fitness poisson

```

```

modelBV_RR_pois_aprilmay <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois_aprilmay,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

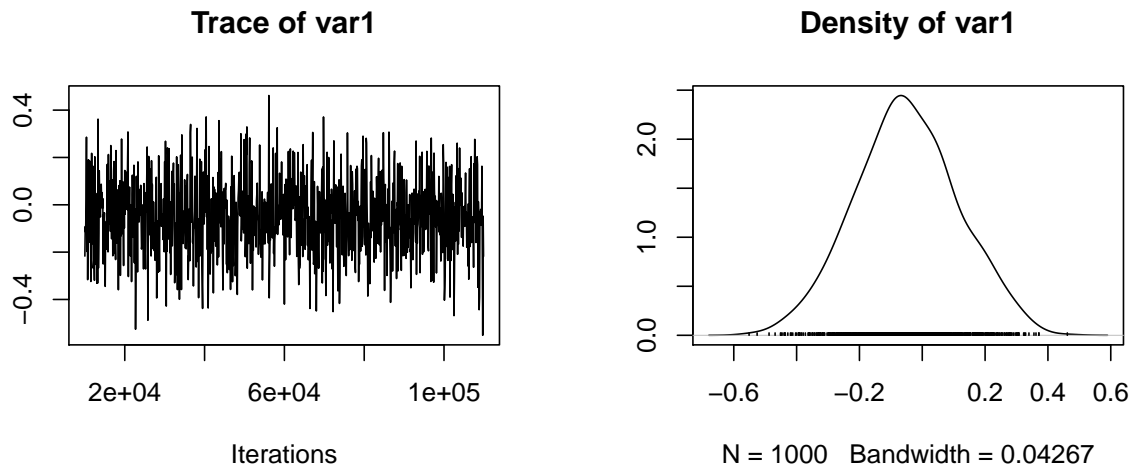
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois_aprilmay_slopefit <-
  modelBV_RR_pois_aprilmay$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]
  (sqrt(modelBV_RR_pois_aprilmay$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\"):temp.id"]
  sqrt(modelBV_RR_pois_aprilmay$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]
  plot(cor_BV_RR_pois_aprilmay_slopefit)

```



```
posterior.mode(cor_BV_RR_pois_aprilmay_slopefit)
```

```
##          var1
## -0.07820992
```

```
HPDinterval(cor_BV_RR_pois_aprilmay_slopefit)
```

```
##          lower      upper
## var1 -0.3618492 0.2779801
## attr(,"Probability")
## [1] 0.95
```

Mean fitness per flowering event

Using mean fitness per flowering event (sum of fitness divided by number of years when the plant flowered), instead of mean fitness over all years.

```
data_4yrs_total <- data_4yrs_total %>%
  right_join(data_4yrs %>%
    group_by(id) %>%
    summarise(n_years = n(), mean_fitness_fl = sum(n_intact_seeds) / mean(n_years)) %>%
    select(id, mean_fitness_fl), by = "id")
```

Temperature: mean April

Without number of flowers

Stack data:

```
# Create a single data-set "data.stack_pois2", with single column at start to index observations
data.stack_pois2 <- c()
data.stack_pois2$Obs <- 1:(243 + 1455)
data.stack_pois2$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2$year <- c(data_4yrs_total$first_yr,
  data_4yrs$year)
```

```

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2$temp <- c(rep(0, 243), data_4yrs$cmean_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness_fl), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois2$variable <- data.stack_pois2$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois2$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois2 <- data.frame(data.stack_pois2)

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

```

```

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1 1 2006    0                14 fitness  fitness poisson
## 2    2 2 2007    0                 4 fitness  fitness poisson
## 3    3 3 2007    0                 2 fitness  fitness poisson
## 4    4 4 2007    0                 6 fitness  fitness poisson
## 5    5 5 2007    0                 4 fitness  fitness poisson
## 6    6 6 2007    0                 2 fitness  fitness poisson

```

```

modelBV_RR_pois2 <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

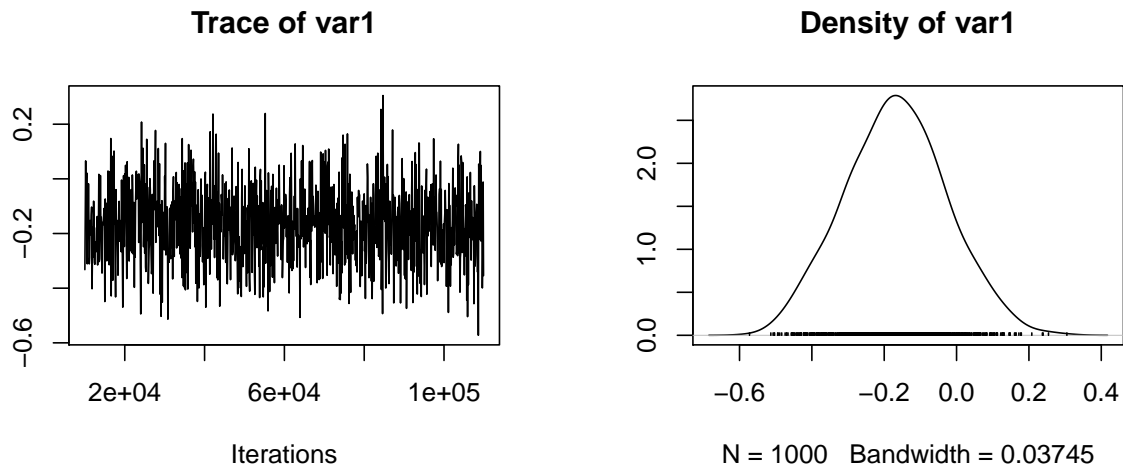
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois2_slopefit <-
  modelBV_RR_pois2$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois2$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois2$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois2_slopefit)

```



```
posterior.mode(cor_BV_RR_pois2_slopefit)
```

```
##      var1
## -0.1766855
```

```
HPDinterval(cor_BV_RR_pois2_slopefit)
```

```
##           lower      upper
## var1 -0.4343329 0.1130045
## attr("Probability")
## [1] 0.95
```

With number of flowers

Stack data:

```
# Create a single data-set "data.stack_pois2_nfl",
# with single column at start to index observations
data.stack_pois2_nfl <- c()
data.stack_pois2_nfl$Obs <- 1:(243 + 1455)
data.stack_pois2_nfl$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2_nfl$year <- c(data_4yrs_total_wfirstyr$first_yr,
                               data_4yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2_nfl$temp <- c(rep(0, 243), data_4yrs$cmean_4)

# Number of flowers column is only relevant for FFD, but is set to 0 for fitness values
data_4yrs$cn_fl <- scale(data_4yrs$n_fl, center=T, scale=F) # Standardize number of flowers
data.stack_pois2_nfl$n_fl <- c(rep(0, 243), data_4yrs$cn_fl)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2_nfl$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness_fl), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2_nfl$traits <- c(rep("fitness", 243), rep("FFD", 1455))
```

```

data.stack_pois2_nfl$variable <- data.stack_pois_nfl$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_pois2_nfl$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois2_nfl <- data.frame(data.stack_pois2_nfl)

data.stack_pois2_nfl$id <- as.factor(data.stack_pois2_nfl$id)
data.stack_pois2_nfl$year <- as.factor(data.stack_pois2_nfl$year)
head(data.stack_pois2_nfl)

```

```

##   Obs id year temp n_fl fitness.FFD.stack  traits variable  family
## 1    1  1 2006    0    0          16 fitness  fitness poisson
## 2    2  2 2007    0    0           6 fitness  fitness poisson
## 3    3  3 2007    0    0           3 fitness  fitness poisson
## 4    4  4 2007    0    0           7 fitness  fitness poisson
## 5    5  5 2007    0    0           4 fitness  fitness poisson
## 6    6  6 2007    0    0           3 fitness  fitness poisson

```

```

modelBV_RR_pois2_nfl <- 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, "FFD"):n_fl, # single fixed effect of n_fl
  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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2_nfl,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

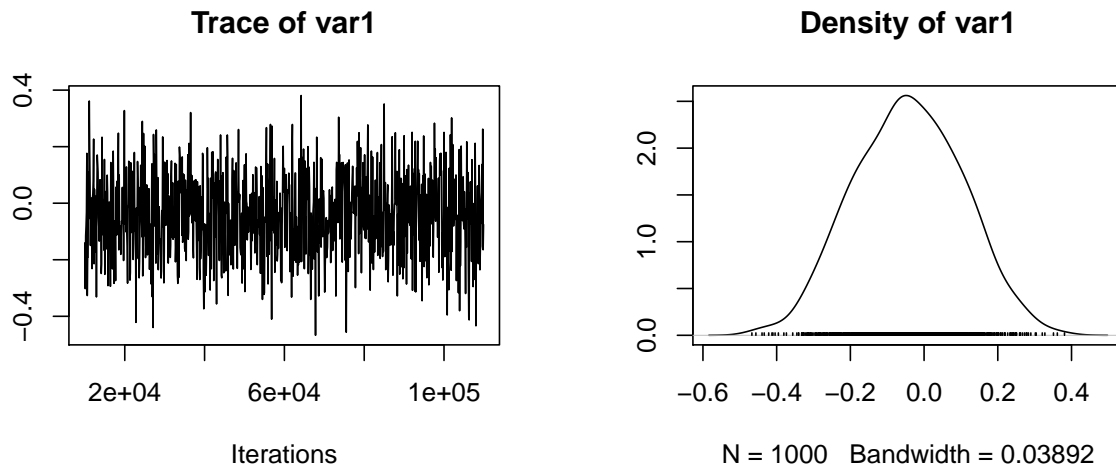
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois2_nfl_slopefit <-
  modelBV_RR_pois2_nfl$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois2_nfl$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois2_nfl$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois2_nfl_slopefit)

```



```
posterior.mode(cor_BV_RR_pois2_nfl_slopefit)
```

```
##      var1
## 0.03757234
```

```
HPDinterval(cor_BV_RR_pois2_nfl_slopefit)
```

```
##           lower      upper
## var1 -0.3322421 0.2195263
## attr("Probability")
## [1] 0.95
```

Temperature: mean May

Stack data:

```
# Create a single data-set "data.stack_pois2_may", with single column at start to index observations
data.stack_pois2_may <- c()
data.stack_pois2_may$Obs <- 1:(243 + 1455)
data.stack_pois2_may$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2_may$year <- c(data_4yrs_total$first_yr,
                               data_4yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2_may$temp <- c(rep(0, 243), data_4yrs$cmean_5)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2_may$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness_fl), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2_may$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois2_may$variable <- data.stack_pois2_may$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois2_may$family <- c(rep("poisson", 243), rep("gaussian", 1455))
```



```

data.stack_pois2_may <- data.frame(data.stack_pois2_may)

data.stack_pois2_may$id <- as.factor(data.stack_pois2_may$id)
data.stack_pois2_may$year <- as.factor(data.stack_pois2_may$year)
head(data.stack_pois2_may)

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0          16 fitness  fitness poisson
## 2    2  2 2007    0           6 fitness  fitness poisson
## 3    3  3 2007    0           3 fitness  fitness poisson
## 4    4  4 2007    0           7 fitness  fitness poisson
## 5    5  5 2007    0           4 fitness  fitness poisson
## 6    6  6 2007    0           3 fitness  fitness poisson

modelBV_RR_pois2_may <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2_may,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

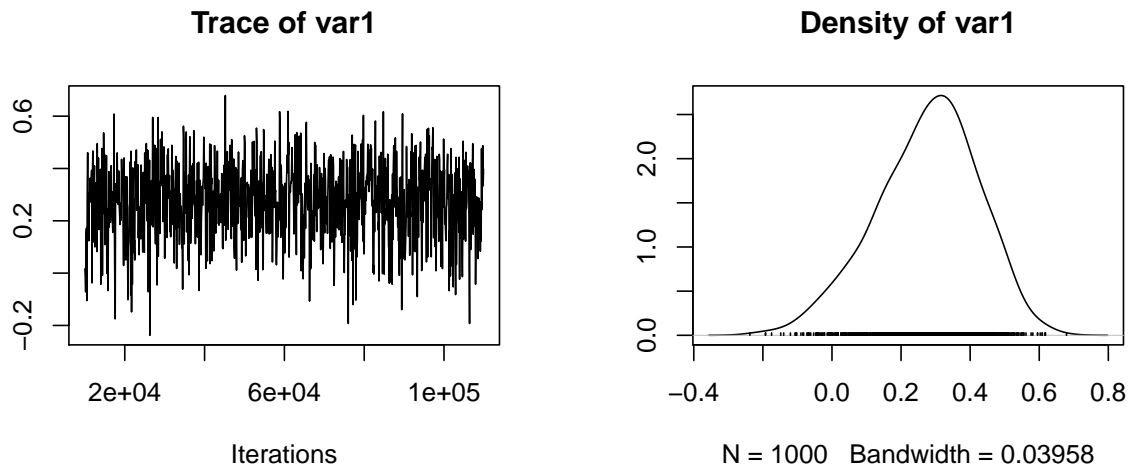
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois2_may_slopefit <-
  modelBV_RR_pois2_may$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois2_may$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois2_may$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois2_may_slopefit)

```



```
posterior.mode(cor_BV_RR_pois2_may_slopefit)
```

```
##      var1
## 0.3352356
```

```
HPDinterval(cor_BV_RR_pois2_may_slopefit)
```

```
##           lower      upper
## var1 -0.0138436 0.555845
## attr("Probability")
## [1] 0.95
```

Temperature: min April

Stack data:

```
# Create a single data-set "data.stack_pois2_min", with single column at start to index observations
data.stack_pois2_min <- c()
data.stack_pois2_min$Obs <- 1:(243 + 1455)
data.stack_pois2_min$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2_min$year <- c(data_4yrs_total$first_yr,
                               data_4yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2_min$temp <- c(rep(0, 243), data_4yrs$cmin_4)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2_min$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness_fl), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2_min$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois2_min$variable <- data.stack_pois2_min$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois2_min$family <- c(rep("poisson", 243), rep("gaussian", 1455))
```

```

data.stack_pois2_min <- data.frame(data.stack_pois2_min)

data.stack_pois2_min$id <- as.factor(data.stack_pois2_min$id)
data.stack_pois2_min$year <- as.factor(data.stack_pois2_min$year)
head(data.stack_pois2_min)

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0          16 fitness  fitness poisson
## 2    2  2 2007    0           6 fitness  fitness poisson
## 3    3  3 2007    0           3 fitness  fitness poisson
## 4    4  4 2007    0           7 fitness  fitness poisson
## 5    5  5 2007    0           4 fitness  fitness poisson
## 6    6  6 2007    0           3 fitness  fitness poisson

modelBV_RR_pois2_min <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2_min,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

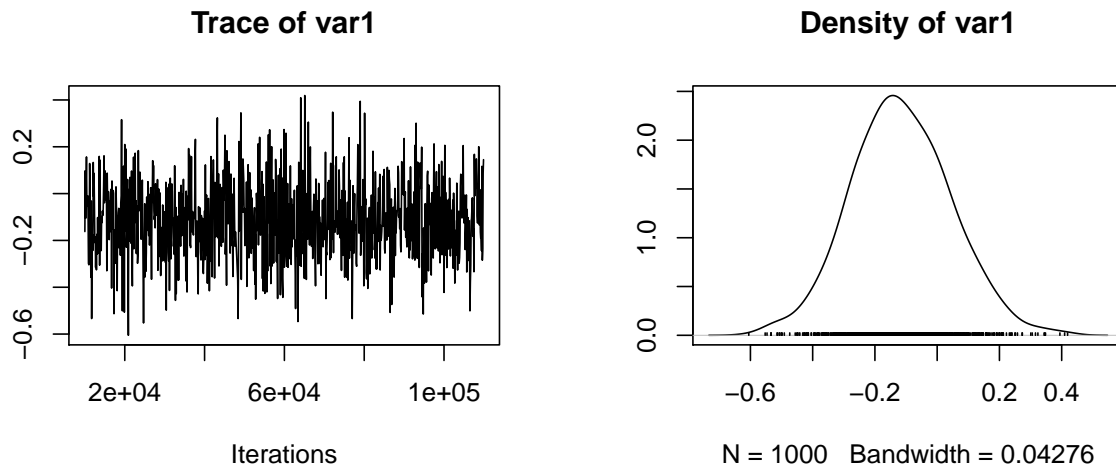
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois2_min_slopefit <-
  modelBV_RR_pois2_min$VCV["at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois2_min$VCV["at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois2_min$VCV["at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois2_min_slopefit)

```



```
posterior.mode(cor_BV_RR_pois2_min_slopefit)
```

```
##          var1
## -0.09822202
```

```
HPDinterval(cor_BV_RR_pois2_min_slopefit)
```

```
##          lower      upper
## var1 -0.4151616 0.2124247
## attr("Probability")
## [1] 0.95
```

Temperature: mean April-May

Stack data:

```
# Create a single data-set "data.stack_pois2_aprilmay",
# with single column at start to index observations
data.stack_pois2_aprilmay <- c()
data.stack_pois2_aprilmay$Obs <- 1:(243 + 1455)
data.stack_pois2_aprilmay$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2_aprilmay$year <- c(data_4yrs_total_wfirstyr$first_yr,
                                     data_4yrs$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2_aprilmay$temp <- c(rep(0, 243), data_4yrs$cmean_45)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2_aprilmay$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness_fl), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2_aprilmay$traits <- c(rep("fitness", 243), rep("FFD", 1455))
data.stack_pois2_aprilmay$variable <- data.stack_pois2_aprilmay$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_pois2_aprilmay$family <- c(rep("poisson", 243), rep("gaussian", 1455))
data.stack_pois2_aprilmay <- data.frame(data.stack_pois2_aprilmay)
```

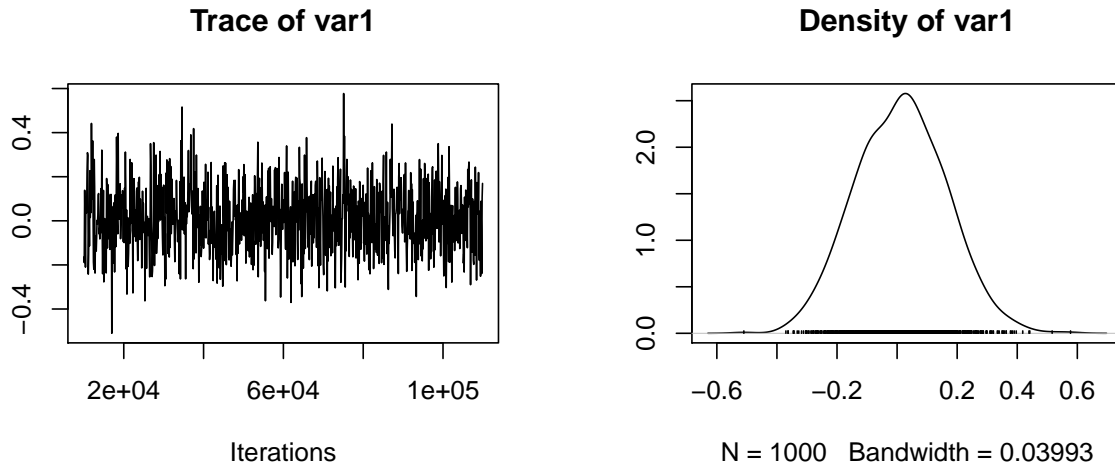
```
data.stack_pois2_aprilmay$id <- as.factor(data.stack_pois2_aprilmay$id)
data.stack_pois2_aprilmay$year <- as.factor(data.stack_pois2_aprilmay$year)
head(data.stack_pois2_aprilmay)
```

```
##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0          16 fitness  fitness poisson
## 2    2  2 2007    0           6 fitness  fitness poisson
## 3    3  3 2007    0           3 fitness  fitness poisson
## 4    4  4 2007    0           7 fitness  fitness poisson
## 5    5  5 2007    0           4 fitness  fitness poisson
## 6    6  6 2007    0           3 fitness  fitness poisson
```

```
modelBV_RR_pois2_aprilmay <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2_aprilmay,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```
cor_BV_RR_pois2_aprilmay_slopefit <-
  modelBV_RR_pois2_aprilmay$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois2_aprilmay$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois2_aprilmay$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\").id"])))
plot(cor_BV_RR_pois2_aprilmay_slopefit)
```



```
posterior.mode(cor_BV_RR_pois2_aprilmay_slopefit)
```

```
##      var1
## 0.02172168
```

```
HPDinterval(cor_BV_RR_pois2_aprilmay_slopefit)
```

```
##      lower      upper
## var1 -0.2724656 0.2983329
## attr(,"Probability")
## [1] 0.95
```

Summary 4 years

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

```
cor_slopefit<-rbind(cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_nfl_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_nfl_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_may_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_may_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_min_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_min_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_aprilmay_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_aprilmay_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_nfl_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_nfl_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_may_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_may_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_min_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_min_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_aprilmay_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_aprilmay_slopefit)[,1:2])))
row.names(cor_slopefit)<-c("Total fitness, mean April temperature, without nfl",
```

```

      "Total fitness, mean April temperature, with n_fl",
      "Total fitness, mean May temperature",
      "Total fitness, min April temperature",
      "Total fitness, mean April-May temperature",
      "Mean fitness per flowering event, mean April temperature, without n_fl",
      "Mean fitness per flowering event, mean April temperature, with n_fl",
      "Mean fitness per flowering event, mean May temperature",
      "Mean fitness per flowering event, min April temperature",
      "Mean fitness per flowering event, mean April-May temperature")
kable(cor_slopefit,digits=3)

```

	posterior.mode	lower	upper
Total fitness, mean April temperature, without n_fl	-0.225	-0.501	0.067
Total fitness, mean April temperature, with n_fl	-0.041	-0.350	0.238
Total fitness, mean May temperature	0.280	-0.010	0.591
Total fitness, min April temperature	-0.200	-0.514	0.126
Total fitness, mean April-May temperature	-0.078	-0.362	0.278
Mean fitness per flowering event, mean April temperature, without n_fl	-0.177	-0.434	0.113
Mean fitness per flowering event, mean April temperature, with n_fl	0.038	-0.332	0.220
Mean fitness per flowering event, mean May temperature	0.335	-0.014	0.556
Mean fitness per flowering event, min April temperature	-0.098	-0.415	0.212
Mean fitness per flowering event, mean April-May temperature	0.022	-0.272	0.298

Using ids with 5 years of data

All models without number of flowers.

Total fitness

Temperature: mean April

Without number of flowers

```

# Create a single data-set "data.stack_pois_5", with single column at start to index observations
data.stack_pois_5 <- c()
data.stack_pois_5$Obs <- 1:(156 + 1107)
data.stack_pois_5$id <- c(data_5yrs_total$id, data_5yrs$id)

# 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_wfirstyr<-data_5yrs_total%>%
  right_join(data_5yrs[c(3,10)]%>%
    group_by(id)%>%
    summarise(first_yr=mean(first_yr)),by="id")

data.stack_pois_5$year <- c(data_5yrs_total_wfirstyr$first_yr,
  data_5yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_5$temp <- c(rep(0, 156), data_5yrs$cmean_4)

```

```

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_5$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois_5$traits <- c(rep("fitness", 156), rep("FFD", 1107))
data.stack_pois_5$variable <- data.stack_pois_5$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois_5$family <- c(rep("poisson", 156), rep("gaussian", 1107))
data.stack_pois_5 <- data.frame(data.stack_pois_5)

data.stack_pois_5$id <- as.factor(data.stack_pois_5$id)
data.stack_pois_5$year <- as.factor(data.stack_pois_5$year)
head(data.stack_pois_5)

```

```

##   Obs id year temp fitness.FFD.stack  traits variable family
## 1    1  1 2006    0              14 fitness  fitness poisson
## 2    2  2 2007    0               4 fitness  fitness poisson
## 3    3  3 2007    0               2 fitness  fitness poisson
## 4    4  4 2007    0               6 fitness  fitness poisson
## 5    5  5 2007    0               4 fitness  fitness poisson
## 6    6  6 2007    0               2 fitness  fitness poisson

```

```

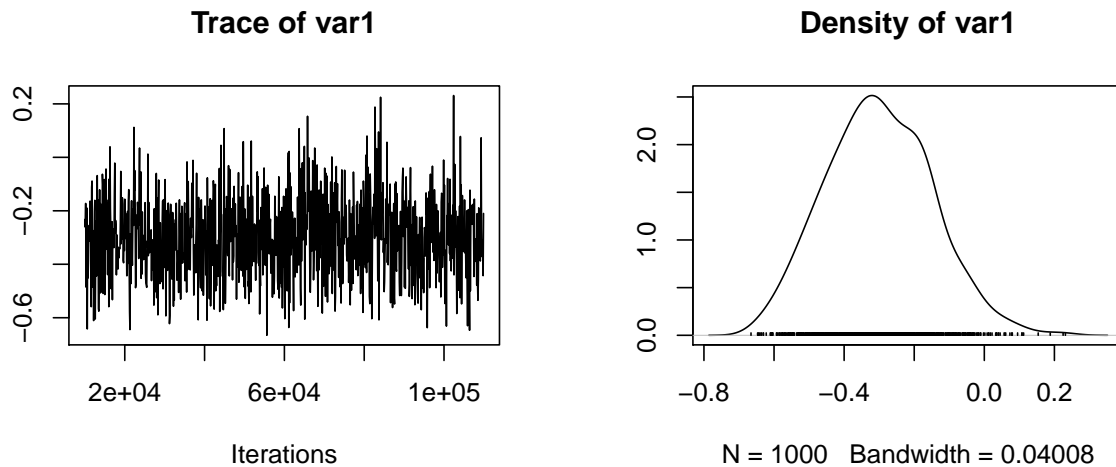
modelBV_RR_pois_5 <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
  # ^ means for each variable (and no overall mean (hence "-1"))
  at.level(variable, "FFD"):temp, # single fixed effect of temp
  random = ~us(at.level(variable, "FFD")):year +
  us(at.level(variable, "FFD") +
    at.level(variable, "FFD"):temp):id,
  # ^ random intercepts for individual,
  # and random slopes for temp/id
  rcov = ~us(at.level(variable, "fitness")):id +
  # ^ variance between individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois_5,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

```

```

cor_BV_RR_pois_5_slopefit <-
  modelBV_RR_pois_5$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"])
  sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])
plot(cor_BV_RR_pois_5_slopefit)

```

```
posterior.mode(cor_BV_RR_pois_5_slopefit)
```

```
##      var1
## -0.3563453
```

```
HPDinterval(cor_BV_RR_pois_5_slopefit)
```

```
##           lower           upper
## var1 -0.5893757 -0.01694865
## attr("Probability")
## [1] 0.95
```

With number of flowers

Stack data:

```
# Create a single data-set "data.stack_pois_5_nfl",
# with single column at start to index observations
data.stack_pois_5_nfl <- c()
data.stack_pois_5_nfl$Obs <- 1:(243 + 1455)
data.stack_pois_5_nfl$id <- c(data_4yrs_total$id, data_4yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois_5_nfl$year <- c(data_4yrs_total$firstyr$first_yr,
                                data_4yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_5_nfl$temp <- c(rep(0, 243), data_4yrs$cmean_4)

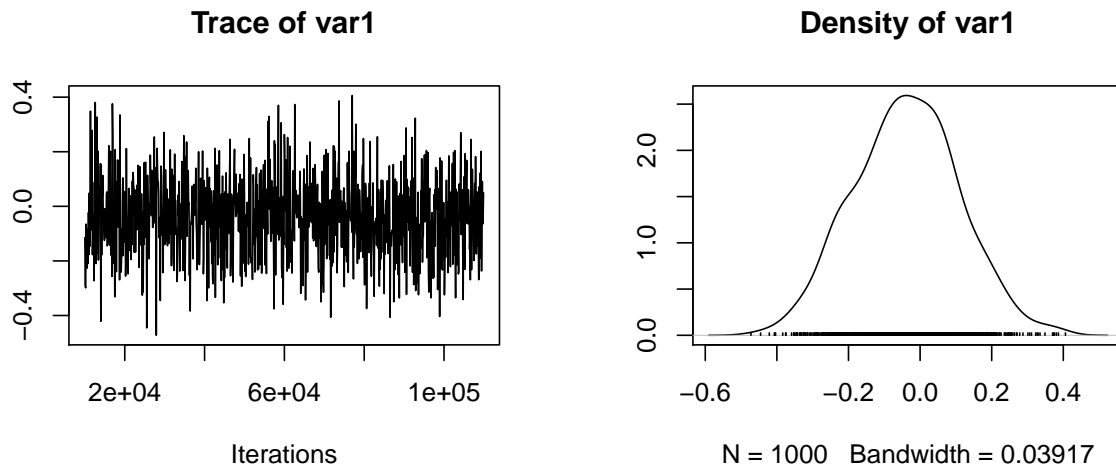
# Number of flowers column is only relevant for FFD, but is set to 0 for fitness values
data_4yrs$cn_fl <- scale(data_4yrs$cn_fl, center=T, scale=F) # Standardize number of flowers
data.stack_pois_5_nfl$cn_fl <- c(rep(0, 243), data_4yrs$cn_fl)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_5_nfl$fitness.FFD.stack <- c(round(data_4yrs_total$mean_fitness), data_4yrs$FFD)

# Create 3 index columns needed for MCMCglmm
```

##	Obs	id	year	temp	n_fl	fitness.FFD.stack	traits	variable	family
##	1	1	1	2006	0	0	14	fitness	fitness poisson
##	2	2	2	2007	0	0	4	fitness	fitness poisson
##	3	3	3	2007	0	0	2	fitness	fitness poisson
##	4	4	4	2007	0	0	6	fitness	fitness poisson
##	5	5	5	2007	0	0	4	fitness	fitness poisson
##	6	6	6	2007	0	0	2	fitness	fitness poisson

```
cor_BV_RR_pois_5_nfl_slopefit <-
  modelBV_RR_pois_5_nfl$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois_5_nfl$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id:temp.id"]) *
   sqrt(modelBV_RR_pois_5_nfl$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"]))
plot(cor_BV_RR_pois_5_nfl_slopefit)
```



```
posterior.mode(cor_BV_RR_pois_5_nfl_slopefit)
```

```
##          var1
## 0.009830252
```

```
HPDinterval(cor_BV_RR_pois_5_nfl_slopefit)
```

```
##          lower      upper
## var1 -0.3534902 0.2128344
## attr("Probability")
## [1] 0.95
```

Temperature: mean May

Mean-centred mean May temperature:

```
data_5yrs<-data_5yrs%>%
  mutate(cmean_5=scale(mean_5,center=T,scale=F))
```

Stack data:

```
# Create a single data-set "data.stack_pois_5_may", with single column at start to index observations
data.stack_pois_5_may <- c()
data.stack_pois_5_may$Obs <- 1:(156 + 1107)
data.stack_pois_5_may$id <- c(data_5yrs_total$id, data_5yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois_5_may$year <- c(data_5yrs_total_wfirstyr$first_yr,
                                data_5yrs$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois_5_may$temp <- c(rep(0, 156), data_5yrs$cmean_5)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois_5_may$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois_5_may$traits <- c(rep("fitness", 156), rep("FFD", 1107))
data.stack_pois_5_may$variable <- data.stack_pois_5_may$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_pois_5_may$family <- c(rep("poisson", 156), rep("gaussian", 1107))
data.stack_pois_5_may <- data.frame(data.stack_pois_5_may)

```

```

data.stack_pois_5_may$id <- as.factor(data.stack_pois_5_may$id)
data.stack_pois_5_may$year <- as.factor(data.stack_pois_5_may$year)
head(data.stack_pois_5_may)

```

```

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0          14 fitness  fitness poisson
## 2    2  2 2007    0           4 fitness  fitness poisson
## 3    3  3 2007    0           2 fitness  fitness poisson
## 4    4  4 2007    0           6 fitness  fitness poisson
## 5    5  5 2007    0           4 fitness  fitness poisson
## 6    6  6 2007    0           2 fitness  fitness poisson

```

```

modelBV_RR_pois_5_may <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois_5_may,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

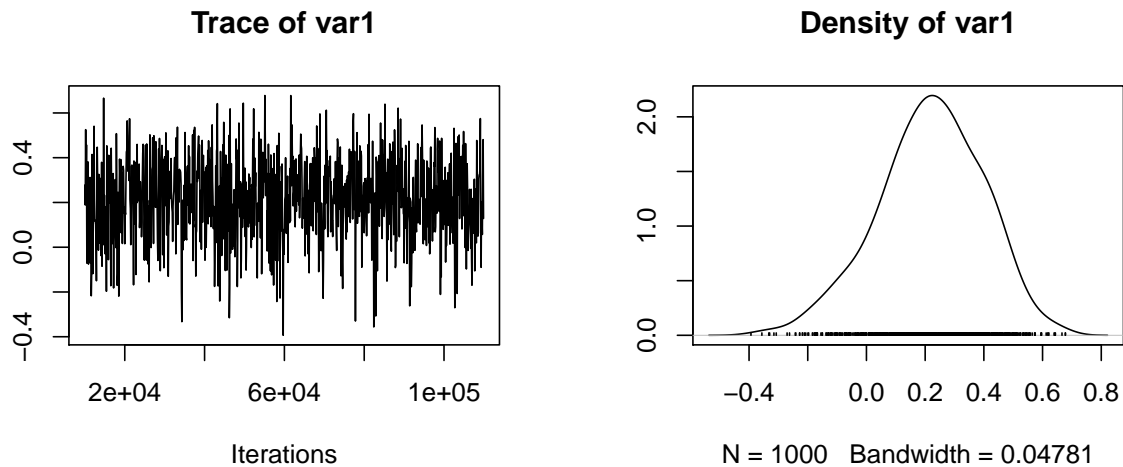
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois_5_may_slopefit <-
  modelBV_RR_pois_5_may$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois_5_may$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois_5_may$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois_5_may_slopefit)

```



```
posterior.mode(cor_BV_RR_pois_5_may_slopefit)
```

```
##      var1
## 0.2204907
```

```
HPDinterval(cor_BV_RR_pois_5_may_slopefit)
```

```
##      lower      upper
## var1 -0.137329 0.5646195
## attr("Probability")
## [1] 0.95
```

Mean fitness per flowering event

```
data_5yrs_total <- data_5yrs_total %>%
  right_join(data_5yrs %>%
    group_by(id) %>%
    summarise(n_years = n(), mean_fitness_fl = sum(n_intact_seeds) / mean(n_years)) %>%
    select(id, mean_fitness_fl, by = "id"))
```

Temperature: mean April

Stack data:

```
# Create a single data-set "data.stack_pois2_5", with single column at start to index observations
data.stack_pois2_5 <- c()
data.stack_pois2_5$obs <- 1:(156 + 1107)
data.stack_pois2_5$id <- c(data_5yrs_total$id, data_5yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2_5$year <- c(data_5yrs_total$first_yr,
                             data_5yrs$year)

# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2_5$temp <- c(rep(0, 156), data_5yrs$cmean_4)
```

```

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2_5$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness_fl), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2_5$traits <- c(rep("fitness", 156), rep("FFD", 1107))
data.stack_pois2_5$variable <- data.stack_pois2_5$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois2_5$family <- c(rep("poisson", 156), rep("gaussian", 1107))
data.stack_pois2_5 <- data.frame(data.stack_pois2_5)

data.stack_pois2_5$id <- as.factor(data.stack_pois2_5$id)
data.stack_pois2_5$year <- as.factor(data.stack_pois2_5$year)
head(data.stack_pois2_5)

```

```

##   Obs id year temp fitness.FFD.stack traits variable family
## 1   1  1 2006    0              16 fitness  fitness poisson
## 2   2  2 2007    0               6 fitness  fitness poisson
## 3   3  3 2007    0               3 fitness  fitness poisson
## 4   4  4 2007    0               7 fitness  fitness poisson
## 5   5  5 2007    0               4 fitness  fitness poisson
## 6   6  6 2007    0               3 fitness  fitness poisson

```

```

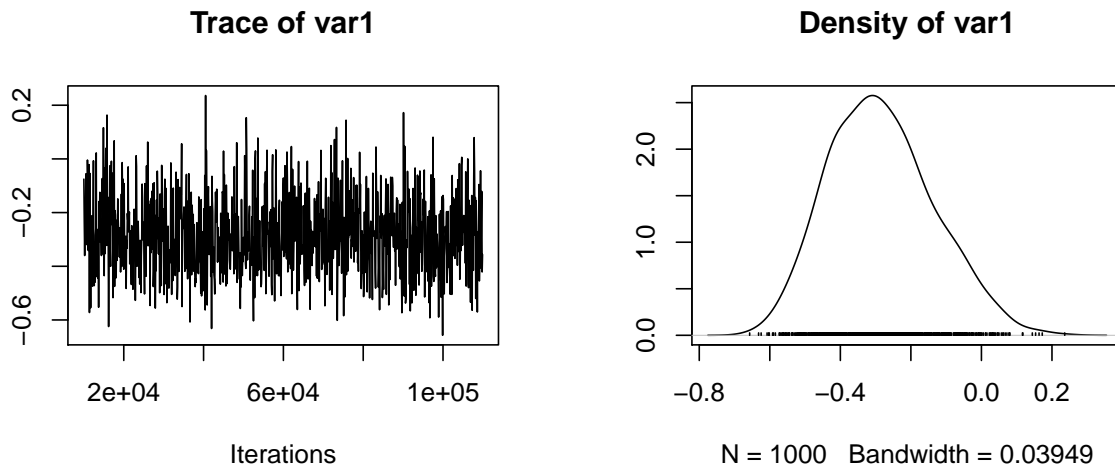
modelBV_RR_pois2_5 <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
  # ^ means for each variable (and no overall mean (hence "-1"))
  at.level(variable, "FFD"):temp, # single fixed effect of temp
  random = ~us(at.level(variable, "FFD")):year +
  us(at.level(variable, "FFD") +
    at.level(variable, "FFD"):temp):id,
  # ^ random intercepts for individual,
  # and random slopes for temp/id
  rcov = ~us(at.level(variable, "fitness")):id +
  # ^ variance between individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2_5,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

```

```

cor_BV_RR_pois2_5_slopefit <-
  modelBV_RR_pois2_5$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois2_5$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois2_5$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois2_5_slopefit)

```



```
posterior.mode(cor_BV_RR_pois2_5_slopefit)
```

```
##      var1
## -0.3239812
```

```
HPDinterval(cor_BV_RR_pois2_5_slopefit)
```

```
##           lower           upper
## var1 -0.5639433 -0.003528689
## attr("Probability")
## [1] 0.95
```

Temperature: mean May

Stack data:

```
# Create a single data-set "data.stack_pois2_5_may", with single column at start to index observations
data.stack_pois2_5_may <- c()
data.stack_pois2_5_may$Obs <- 1:(156 + 1107)
data.stack_pois2_5_may$id <- c(data_5yrs_total$id, data_5yrs$id)

# Year column is only relevant for FFD, but is set to first_yr for fitness values
data.stack_pois2_5_may$year <- c(data_5yrs_total_wfirstyr$first_yr,
                                data_5yrs$year)
# Temperature column is only relevant for FFD, but is set to 0 for fitness values
data.stack_pois2_5_may$temp <- c(rep(0, 156), data_5yrs$cmean_5)

# Create single column with first fitness values (ABSOLUTE VALUES), then FFD values:
data.stack_pois2_5_may$fitness.FFD.stack <- c(round(data_5yrs_total$mean_fitness_fl), data_5yrs$FFD)

# Create 3 index columns needed for MCMCglmm
data.stack_pois2_5_may$traits <- c(rep("fitness", 156), rep("FFD", 1107))
data.stack_pois2_5_may$variable <- data.stack_pois2_5_may$traits
# Fitness will be modelled with an overdispersed Poisson
# FFD will be modelled with a Gaussian distribution
# Specify this with the column 'family':
data.stack_pois2_5_may$family <- c(rep("poisson", 156), rep("gaussian", 1107))
```

```

data.stack_pois2_5_may <- data.frame(data.stack_pois2_5_may)

data.stack_pois2_5_may$id <- as.factor(data.stack_pois2_5_may$id)
data.stack_pois2_5_may$year <- as.factor(data.stack_pois2_5_may$year)
head(data.stack_pois2_5_may)

##   Obs id year temp fitness.FFD.stack traits variable family
## 1    1  1 2006    0          16 fitness  fitness poisson
## 2    2  2 2007    0           6 fitness  fitness poisson
## 3    3  3 2007    0           3 fitness  fitness poisson
## 4    4  4 2007    0           7 fitness  fitness poisson
## 5    5  5 2007    0           4 fitness  fitness poisson
## 6    6  6 2007    0           3 fitness  fitness poisson

modelBV_RR_pois2_5_may <- 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack_pois2_5_may,
  prior = priorBiv_RR_pois,
  family = NULL, # specified already in the data-set
  nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)

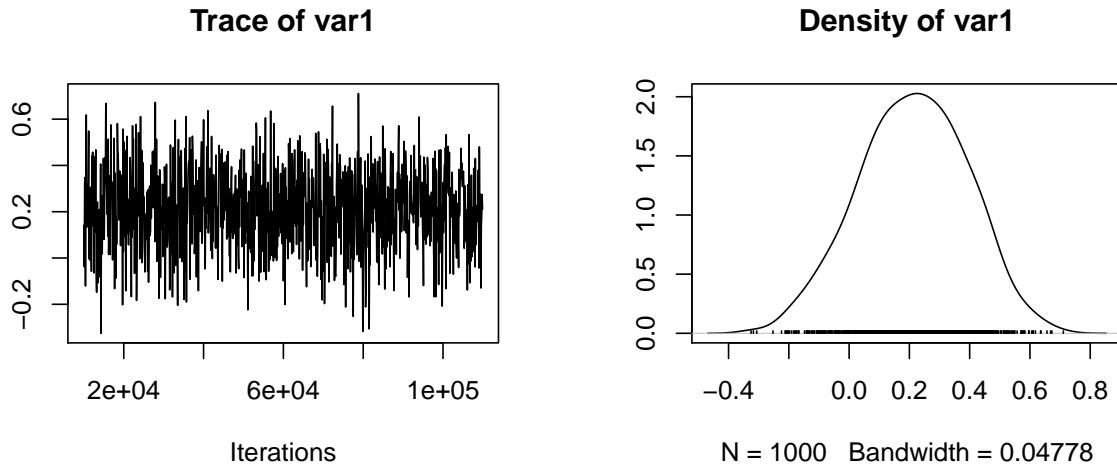
```

Determining the among-individual correlation between fitness and variation in slopes for FFD:

```

cor_BV_RR_pois2_5_may_slopefit <-
  modelBV_RR_pois2_5_may$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"]
  (sqrt(modelBV_RR_pois2_5_may$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\")"]
    sqrt(modelBV_RR_pois2_5_may$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):"]
  plot(cor_BV_RR_pois2_5_may_slopefit)

```

```
posterior.mode(cor_BV_RR_pois2_5_may_slopefit)
```

```
##      var1
## 0.2186931
```

```
HPDinterval(cor_BV_RR_pois2_5_may_slopefit)
```

```
##           lower      upper
## var1 -0.1415346 0.5413856
## attr(,"Probability")
## [1] 0.95
```

Summary 5 years

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

```
cor_slopefit_5<-rbind(cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_5_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_5_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_5_nfl_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_5_nfl_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois_5_may_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois_5_may_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_5_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_5_slopefit)[,1:2])),
  cbind(posterior.mode=posterior.mode(cor_BV_RR_pois2_5_may_slopefit),
  rbind(HPDinterval(cor_BV_RR_pois2_5_may_slopefit)[,1:2])))
row.names(cor_slopefit_5)<-c("Total fitness, mean April temperature, without n_fl",
  "Total fitness, mean April temperature, with n_fl",
  "Total fitness, mean May temperature",
  "Mean fitness per flowering event, mean April temperature",
  "Mean fitness per flowering event, mean May temperature")
kable(cor_slopefit_5,digits=3)
```

	posterior.mode	lower	upper
Total fitness, mean April temperature, without n_fl	-0.356	-0.589	-0.017
Total fitness, mean April temperature, with n_fl	0.010	-0.353	0.213

	posterior.mode	lower	upper
Total fitness, mean May temperature	0.220	-0.137	0.565
Mean fitness per flowering event, mean April temperature	-0.324	-0.564	-0.004
Mean fitness per flowering event, mean May temperature	0.219	-0.142	0.541

Only model with selection on RN slope: modelBV_RR_pois_5

```
kable(summary(modelBV_RR_pois_5)$solutions,digits=c(3,3,3,0,3),caption="Fixed effects")
```

Table 7: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	57.494	55.160	59.607	1000	0.001
variablefitness	1.211	1.067	1.345	946	0.001
at.level(variable, "FFD"):temp	-2.398	-3.816	-0.733	1147	0.002

```
kable(summary(modelBV_RR_pois_5)$Gcovariances,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"):at.level(variable, "FFD").year	25.802	11.499	43.407	1127

```
kable(summary(modelBV_RR_pois_5)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 9: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, "FFD").id:at.level(variable, "FFD").id	3.405	2.111	4.676	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	1.041	0.541	1.723	1000
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.620	-0.993	-0.305	1000
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	1.041	0.541	1.723	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	0.776	0.350	1.226	1000
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.183	-0.387	-0.005	1000
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.620	-0.993	-0.305	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.183	-0.387	-0.005	1000
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.482	0.327	0.653	959
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	18.740	16.838	20.345	1000

```
kable(diag(autocorr(modelBV_RR_pois_5$VCV)[2, , ]),caption="Autocorrelation")
```

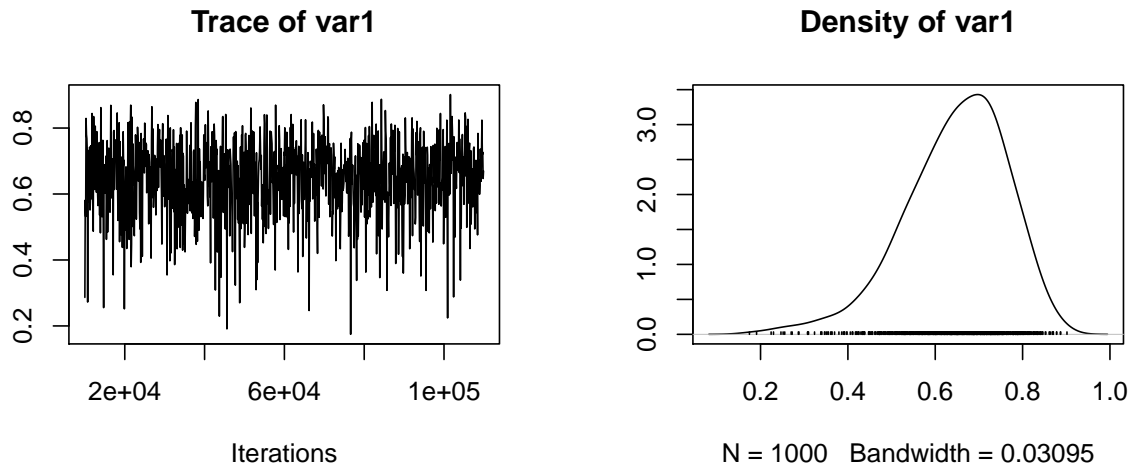
Table 10: Autocorrelation

	x
at.level(variable, "FFD"):at.level(variable, "FFD").year	0.0310838
at.level(variable, "FFD").id:at.level(variable, "FFD").id	-0.0244161
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	-0.0331450

	x
at.level(variable, "fitness").id:at.level(variable, "FFD").id	0.0224512
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	-0.0331450
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	-0.0264357
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.0048342
at.level(variable, "FFD").id:at.level(variable, "fitness").id	0.0224512
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.0048342
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.0433784
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	0.0287729

Among-individual correlation between intercepts and slopes for FFD:

```
cor_BV_RR_pois_5_intslope <-
  modelBV_RR_pois_5$VCV[, "at.level(variable, \"FFD\") :temp.id:at.level(variable, \"FFD\") .id"] /
  (sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"FFD\") .id:at.level(variable, \"FFD\") .id"]) *
  sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"FFD\") :temp.id:at.level(variable, \"FFD\") :temp.id"])))
plot(cor_BV_RR_pois_5_intslope)
```



```
posterior.mode(cor_BV_RR_pois_5_intslope)
```

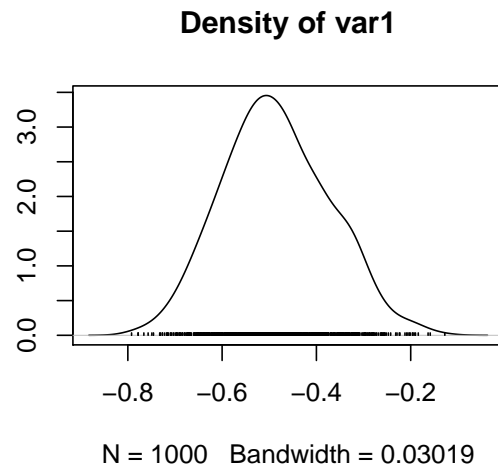
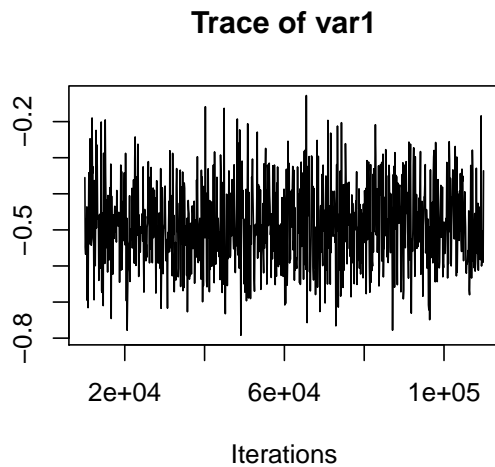
```
##      var1
## 0.6885208
```

```
HPDinterval(cor_BV_RR_pois_5_intslope)
```

```
##      lower      upper
## var1 0.4102019 0.8531525
## attr("Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
cor_BV_RR_pois_5_intfit <-
  modelBV_RR_pois_5$VCV[, "at.level(variable, \"fitness\") .id:at.level(variable, \"FFD\") .id"] /
  (sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"fitness\") .id:at.level(variable, \"fitness\") .id"]) *
  sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"FFD\") .id:at.level(variable, \"FFD\") .id"])))
plot(cor_BV_RR_pois_5_intfit)
```



```
posterior.mode(cor_BV_RR_pois_5_intfit)
```

```
##      var1
## -0.4551948
```

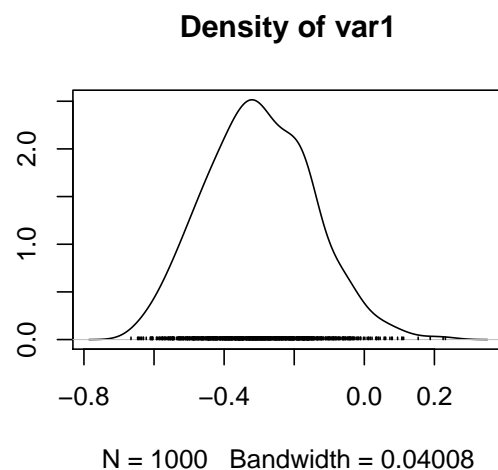
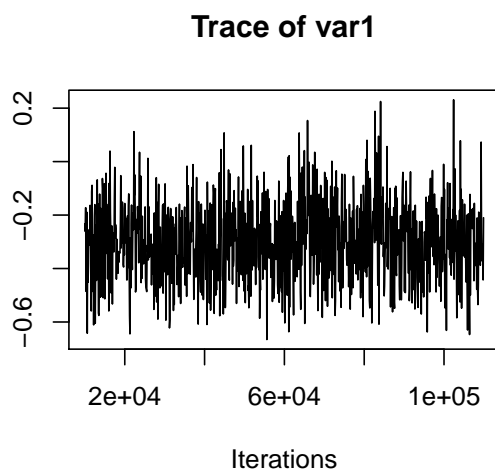
```
HPDinterval(cor_BV_RR_pois_5_intfit)
```

```
##           lower      upper
## var1 -0.7081592 -0.2775473
## attr("Probability")
## [1] 0.95
```

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

```
cor_BV_RR_pois_5_slopefit <-
```

```
  modelBV_RR_pois_5$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"FFD\"):temp.id"] /
  (sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"fitness\").id:at.level(variable, \"fitness\").id"] *
    sqrt(modelBV_RR_pois_5$VCV[, "at.level(variable, \"FFD\"):temp.id:at.level(variable, \"FFD\"):temp.id"])))
plot(cor_BV_RR_pois_5_slopefit)
```



```
posterior.mode(cor_BV_RR_pois_5_slopefit)
```

```
##          var1  
## -0.3563453
```

```
HPDinterval(cor_BV_RR_pois_5_slopefit)
```

```
##          lower          upper  
## var1 -0.5893757 -0.01694865  
## attr("Probability")  
## [1] 0.95
```

Extract selection coefficients

```
# Extract 3x3 matrix of variance-covariance values for intercepts and slopes of X, and LBS  
# These are in the 2nd-10th columns of model output  
P.modelBV_RR_pois_5 <- modelBV_RR_pois_5$VCV[,2:10]  
P.modelBV_RR_pois_5.mode <- matrix(1:9, nrow = 3)  
for (k in 1:9) P.modelBV_RR_pois_5.mode[k] <- posterior.mode(P.modelBV_RR_pois_5[,k])  
P.modelBV_RR_pois_5.mode
```

```
##          [,1]      [,2]      [,3]  
## [1,]  3.5003523  1.0220040 -0.6122982  
## [2,]  1.0220040  0.6412418 -0.1740813  
## [3,] -0.6122982 -0.1740813  0.4853696
```

```
# Extract selection *differentials* (i.e. covariances) for intercept and slope:  
# and calculate posterior mode and credible intervals for each  
S.modelBV_RR_pois_5 <- modelBV_RR_pois_5$VCV[, c(4,7)]  
S.modelBV_RR_pois_5 <- P.modelBV_RR_pois_5[, c(3,6)]  
colnames(S.modelBV_RR_pois_5) <- c("S_intercepts", "S_slopes")  
S.modelBV_RR_pois_5.mode <- P.modelBV_RR_pois_5.mode[1:2, 3]  
S.modelBV_RR_pois_5.mode
```

```
## [1] -0.6122982 -0.1740813
```

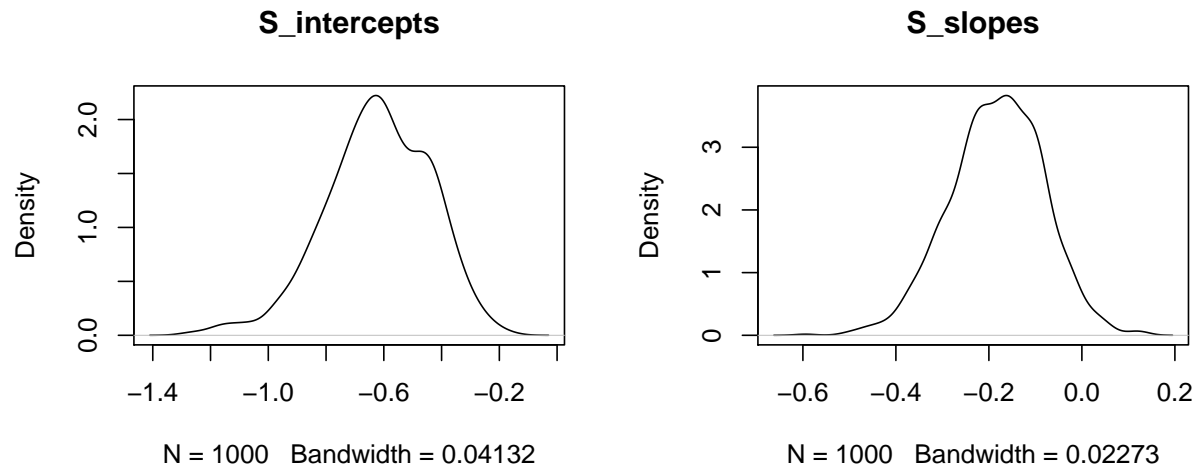
```
posterior.mode(mcmc(S.modelBV_RR_pois_5))
```

```
## S_intercepts    S_slopes  
##   -0.6122982   -0.1740813
```

```
HPDinterval(mcmc(S.modelBV_RR_pois_5))
```

```
##          lower          upper  
## S_intercepts -0.9929710 -0.305195789  
## S_slopes     -0.3867651 -0.005488248  
## attr("Probability")  
## [1] 0.95
```

```
# Plot posterior distribution of selection differentials  
par(mfrow = c(1,2))  
plot(density(S.modelBV_RR_pois_5[,1]), main = "S_intercepts")  
plot(density(S.modelBV_RR_pois_5[,2]), main = "S_slopes")
```



```
# Estimate selection gradients for intercept and slope (beta = S / P)
# on each sample of posterior and extract their mode
n <- length(modelBV_RR_pois_5$VCV[,2]) # sample size
beta_post_RR_pois_5 <- 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_RR_pois_5[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_RR_pois_5[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/

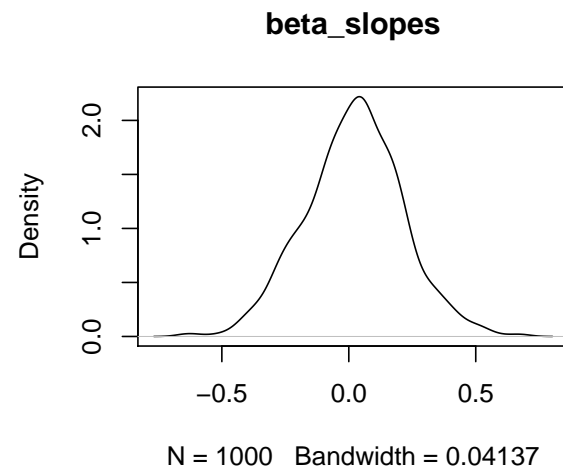
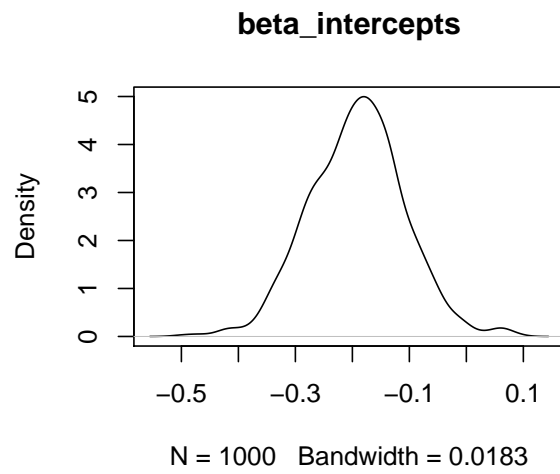
colnames(beta_post_RR_pois_5) <- c("beta_intercepts", "beta_slopes")
posterior.mode(mcmc(beta_post_RR_pois_5))

## beta_intercepts    beta_slopes
##      -0.17559552      0.06080902

HPDinterval(mcmc(beta_post_RR_pois_5))

##                lower      upper
## beta_intercepts -0.3499730 -0.03235167
## beta_slopes     -0.3508229  0.40774204
## attr(,"Probability")
## [1] 0.95

# Plot posterior distribution of selection gradients
par(mfrow = c(1,2))
plot(density(beta_post_RR_pois_5[,1]), main = "beta_intercepts")
plot(density(beta_post_RR_pois_5[,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