Lathyrus ms2: Selection on reaction norms - multivariate modeling for phenotypic selection on plasticity

Contents

```r
# function to calculate vif for MCMCglmm
# Taken from https://github.com/aufrank/R-hacks/blob/master/MCMCglmm-utils.R#L2
vif.MCMCglmm <- function (fit, intercept.columns = c(1)) {
    nF <- fit$Fixed$nfl
    v <- cov(as.matrix(fit$X[,1:nF]))
    nam <- colnames(fit$Sol[,1:nF])

    v <- v[-intercept.columns, -intercept.columns, drop = FALSE]
    nam <- nam[-intercept.columns]

    d <- diag(v)^0.5
    v <- diag(solve(v/(d %o% d)))
    names(v) <- nam
    v
}
```

Data preparation

The data (file "data_5yrs") contains information on 1162 flowering events from 163 different individuals of Lathyrus vernus during two periods (1987-1996 and 2006-2017). This is a subset of a larger data set from which we selected individuals that had at least 5 years of data on first flowering date, in order to be able to accurately estimate reaction norms.

```r
# Read data
data_5yrs<-read.csv(
  "C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/data/data_5yrs.csv",
  header=T,sep=",",dec=".")

# Have a look at the data
head(data_5yrs)
```

```
##   year     id      FFD n_fl shoot_vol n_intact_seeds   mean_4 n_years_FFD
## 1 2006 new_10 58.27431  180  21975.75    43.48453608 4.611667          11
## 2 2007 new_10 43.03194  290  19662.95     7.00000000 7.203333          11
## 3 2008 new_10 44.78889  156  28529.38    89.48571429 6.656667          11
## 4 2009 new_10 49.54653  180   1000.85     0.00000000 7.186667          11
## 5 2010 new_10 57.30417   28   4885.60     0.04532821 5.285000          11
## 6 2011 new_10 47.06181   75   9160.88     0.29757214 8.438333          11
##   n_years_fl n_years_life
## 1         11           12
## 2         11           12
## 3         11           12
## 4         11           12
## 5         11           12
## 6         11           12
```

```r
str(data_5yrs)
```

```
## 'data.frame':    1162 obs. of  10 variables:
##  $ year          : int  2006 2007 2008 2009 2010 2011 2012 2013 2014 2016 ...
##  $ id            : chr  "new_10" "new_10" "new_10" "new_10" ...
##  $ FFD           : num  58.3 43 44.8 49.5 57.3 ...
##  $ n_fl          : num  180 290 156 180 28 75 42 24 2 52 ...
```

```
##  $ shoot_vol     : num  21976 19663 28529 1001 4886 ...
##  $ n_intact_seeds: num  43.4845 7 89.4857 0 0.0453 ...
##  $ mean_4        : num  4.61 7.2 6.66 7.19 5.29 ...
##  $ n_years_FFD   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ n_years_fl    : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ n_years_life  : int  12 12 12 12 12 12 12 12 12 12 ...
```

Variables:

- Year
- Id: individual
- FFD: first flowering date (as number of days after vernal equinox, so lower values mean earlier flowering), our trait of interest
- n_fl: number of flowers
- shoot_vol: shoot volume (measure of plant size)
- n_intact_seeds: number of intact (i.e. non predated) seeds, our fitness measure
- mean_4: mean daily temperature of April
- n_years_FFD: Number of years when we have data for FFD
- n_years_fl: Number of years when the plant has flowered (could be larger than n_years_FFD if no data on FFD is available for any of the flowering years)
- n_years_life = Number of years when the plant was alive (plants can skip flowering in some years)

The temperature 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)
```

Univariate MCMCglmm models

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

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

priorUV2 <- list(G = list(G1 = list(V = diag(1), nu = 1),  # for random effect of year
                          G2 = list(V = diag(1), nu = 1)), # for random effect of id
                 R = list(R1 = list(V = diag(1), nu = 2)))
priorUV2_RR <- list(G = list(G1 = list(V = diag(1), nu = 1),  # other random effect (YEAR)
                             G2 = list(V = diag(2), nu = 1)),
                        # ^ 2x2 variance-covariance matrix for var in slopes + intercepts
                    R = list(R1 = list(V = diag(1), nu = 2)))
```

FFD with random effects of year and individual-intercept

```
univar.FFD <- MCMCglmm(FFD ~ cmean_4,
                     random = ~year + id,
                     rcov = ~units,
                     data = data_5yrs,
                     prior = priorUV2,
                     family = "gaussian",
                     nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
```

```
# Aim to store 2000 iterations
save(univar.FFD,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/univar.FFD.R
```

```
summary(univar.FFD)
```

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 6906.746
##
##  G-structure:  ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year     24.85    11.62    41.76     2000
##
##                ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id     3.283    1.808     4.68     2000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     20.32     18.6    22.23     2000
##
##  Location effects: FFD ~ cmean_4
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   57.4159  55.4739  59.7023     2000 <5e-04 ***
## cmean_4       -2.4003  -3.8861  -0.7317     2000  0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Random regression for FFD, including random effects of individual slopes and covariance between intercept and slope

```
univar.FFD_RR <- MCMCglmm(FFD ~ cmean_4,
                        random = ~year + us(1 + cmean_4):id,
                        rcov = ~units,
                        data = data_5yrs,
                        prior = priorUV2_RR,
                        family = "gaussian",
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F,pr=T)
# pr= T saves the posterior distribution of the individual random effects
# (analagous to the BLUP from the LMM)
save(univar.FFD_RR,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/univar.FF
```

```
summary(univar.FFD_RR)
```

```
##
##  Iterations = 100001:2099001
```

```
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 6873.293
##
##  G-structure:  ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year     25.56       12    44.79     2000
##
##                ~us(1 + cmean_4):id
##
##                             post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).id     3.202   1.8127    4.481     2000
## cmean_4:(Intercept).id         1.097   0.5252    1.688     2352
## (Intercept):cmean_4.id         1.097   0.5252    1.688     2352
## cmean_4:cmean_4.id             0.703   0.2920    1.160     3391
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     19.33    17.55    21.06     2000
##
##  Location effects: FFD ~ cmean_4
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   57.3567  55.1462  59.6125     2000 <5e-04 ***
## cmean_4       -2.3815  -3.8554  -0.6712     2000  0.007 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

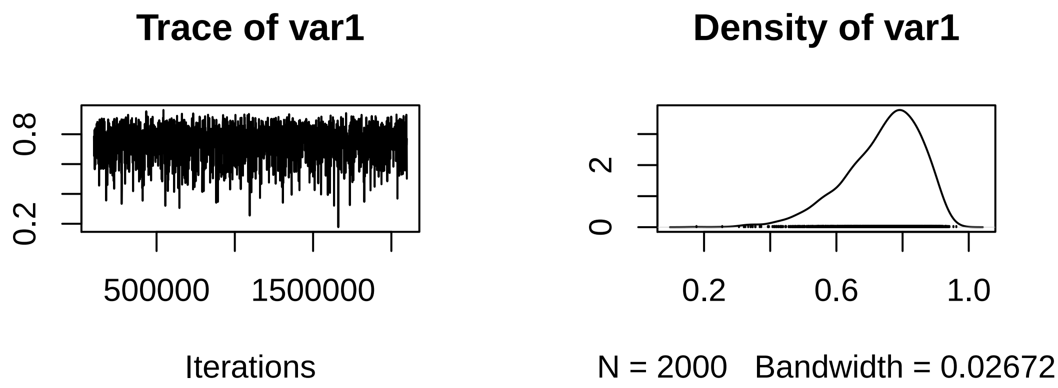
Extract BLUPs from this model

Code adapted from Houslay & Wilson 2017 Behav. Ecol. Code for graphs based on Arnold et al. 2019 Phil. Trans. R. Soc. B.

```
BLUPs_MCMC <- tibble(Trait = attr(colMeans(univar.FFD_RR$Sol), "names"),
                          Value = colMeans(univar.FFD_RR$Sol)) %>%
  filter(grepl("id", Trait))%>% # Select only id intercepts and slopes
  mutate(type=ifelse(grepl("Intercept",Trait),"intercept","slope"))%>%
  mutate(id=sub(".*id.", "", Trait))%>%
  select(-Trait)%>%
  spread(., type, Value) # Convert from long to wide
```

Correlation among intercepts and slopes

```
univar.FFD_RR_intslope <-
  univar.FFD_RR$VCV[,"cmean_4:(Intercept).id"]/
(sqrt(univar.FFD_RR$VCV[,"(Intercept):(Intercept).id"])*
sqrt(univar.FFD_RR$VCV[,"cmean_4:cmean_4.id"]))
plot(univar.FFD_RR_intslope)
```




```r
posterior.mode(univar.FFD_RR_intslope)
```

```
##      var1
## 0.8019481
```

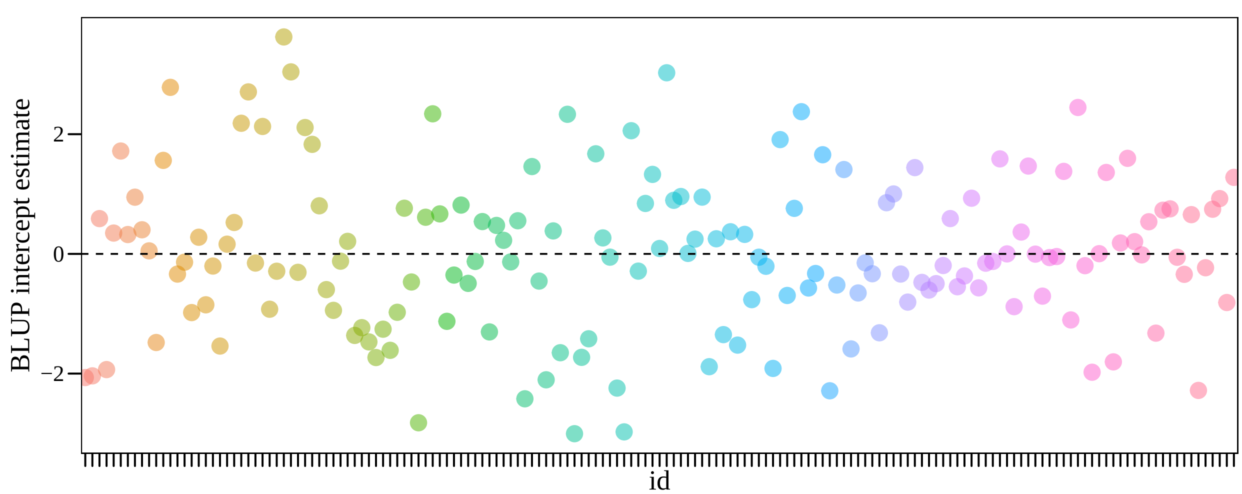
```r
HPDinterval(univar.FFD_RR_intslope)
```

```
##          lower     upper
## var1 0.5175814 0.9323963
## attr(,"Probability")
## [1] 0.95
```

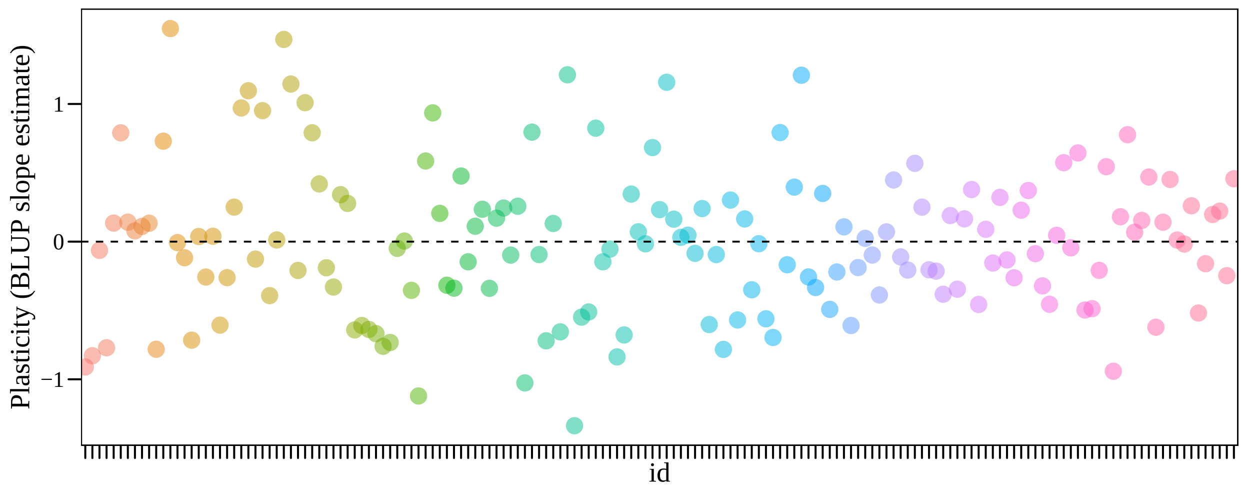
High correlation among BLUPs for intercepts and slopes.

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.

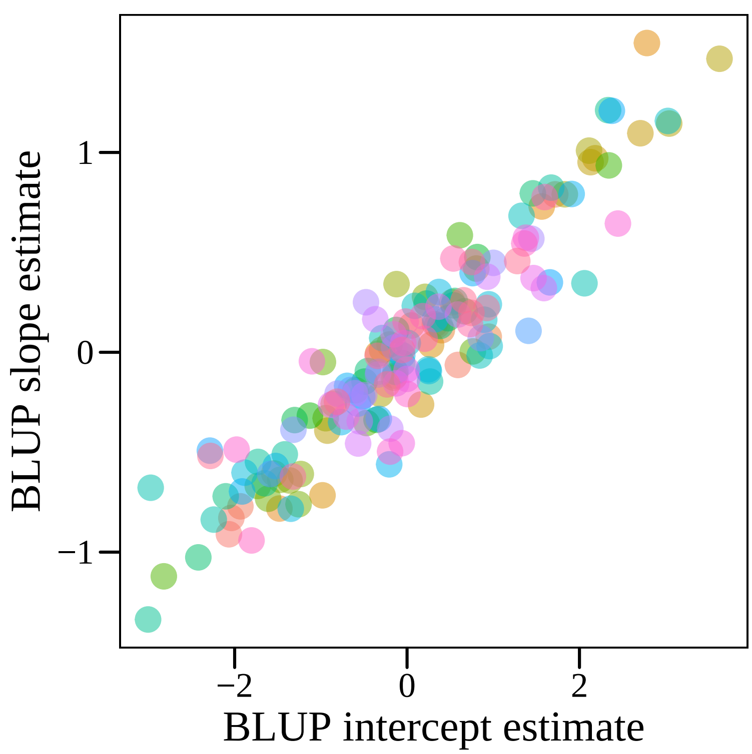
The BLUP intercept term indicates the difference in id elevation relative to the population-average, so more positive values of BLUP intercept indicate that the id's reaction norm occurs above the population-level average and negative values are below the population-level average. The BLUP intercept 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.



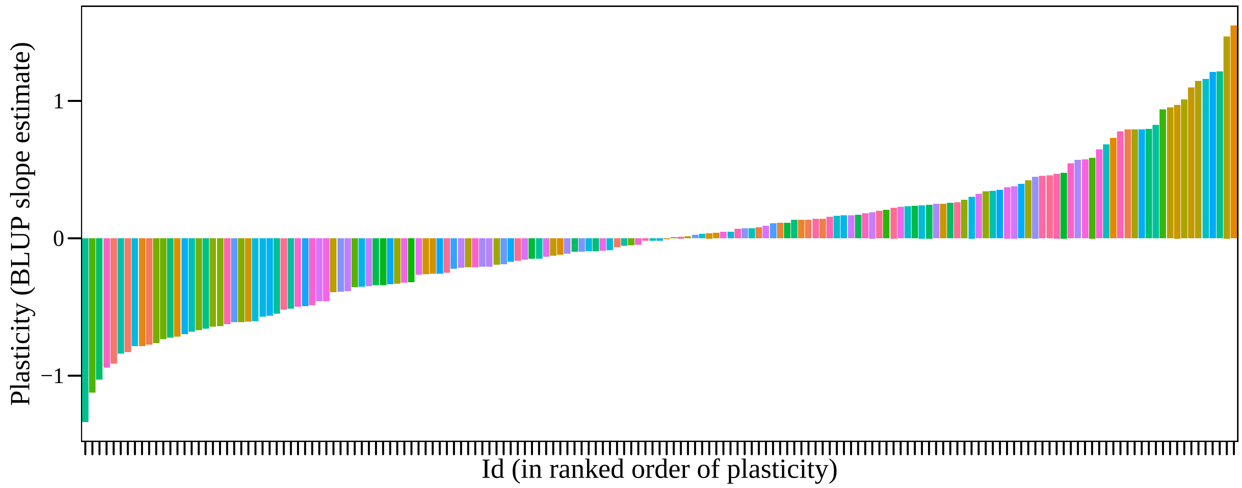
The BLUP slope estimate is the difference in slope (relative steepness of change) between the population-level average response and the response of the id. Here, that is the difference in slope of FFD for each value of temperature relative to the population-level average slope.



The BLUP intercept and slope estimates are highly correlated. 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 have the highest positive intercept, and have the least plasticity across growth temperatures.



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.



Important note! BLUPs are single point estimates that do not have associated measures of uncertainty. Therefore, it is dangerous to derive statistics or make formal interpretation of plasticity based on them without properly accounting for estimation uncertainty.

Bivariate 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 (and using absolute 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 year of life = 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

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

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

```
# Read shoot volume data
shoot_vol<-read.csv(
  "C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/data/shoot_vol.csv",
  header=T,sep=",",dec=".")

# Have a look at shoot volume data
head(shoot_vol)
```

```
##   year     id shoot_vol
## 1 2006 new_10  21975.75
## 2 2007 new_10  19662.95
```

```
## 3 2008 new_10  28529.38
## 4 2009 new_10   1000.85
## 5 2010 new_10   4885.60
## 6 2011 new_10   9160.88
```

```r
str(shoot_vol)
```

```
## 'data.frame':    1310 obs. of  3 variables:
##  $ year     : int  2006 2007 2008 2009 2010 2011 2012 2013 2016 2017 ...
##  $ id       : chr  "new_10" "new_10" "new_10" "new_10" ...
##  $ shoot_vol: num  21976 19663 28529 1001 4886 ...
```

```r
# This file contains values of shoot volume for all ids/years
# (including also yrs when an id was not flowering)

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
##    <chr>               <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
```

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,
```

```r
                    data_5yrs$year)

# 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
```

```r
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 indivduals in fitness
                         # (which is residual variance)
                         us(at.level(variable, "FFD")):Obs,
                         # ^ residual variance within indivduals between years
                       # (labelled by 'Obs')
                       data = data.stack10,
```

```r
                        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/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_RR
```

```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

```r
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

```r
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

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

Table 4: Autocorrelation

	x
variableFFD	0.0147066
variablefitness	0.0102773
at.level(variable, "FFD"):temp	0.0140045

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

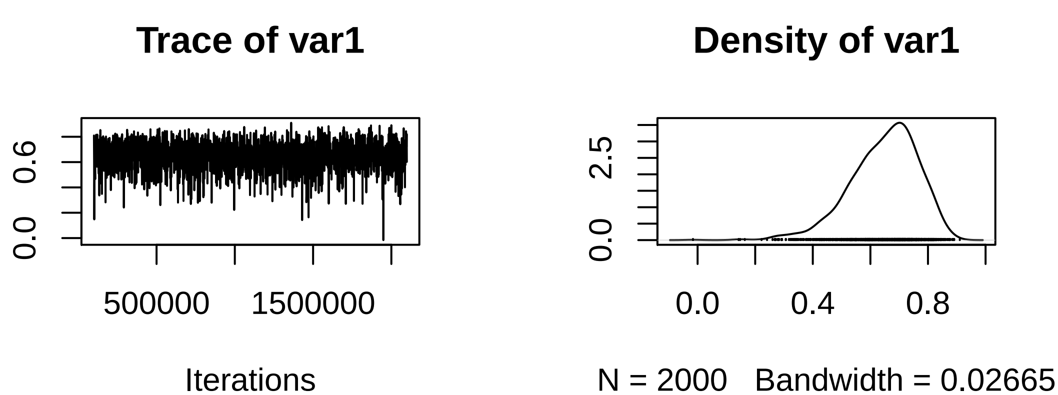
Table 5: Autocorrelation

	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

For interpretation of covariances, we convert them to correlations using the formula for a correlation with the posterior distributions of our (co)variance components. This gives us a distribution of correlation values that we can use to calculate estimates and 95% credible intervals (code adapted from Houslay & Wilson 2017 Behav. Ecol.).

Among-individual correlation between intercepts and slopes for FFD:

```
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)
```

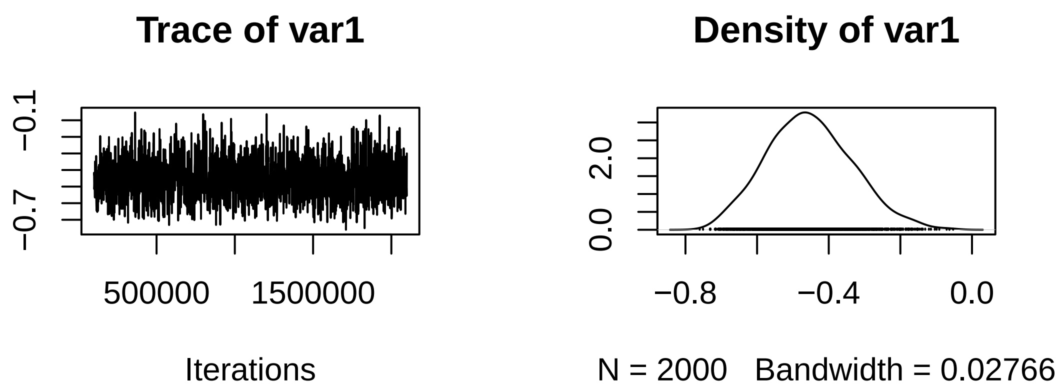
```
##      var1
## 0.7020806
```

```r
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:

```r
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)
```




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

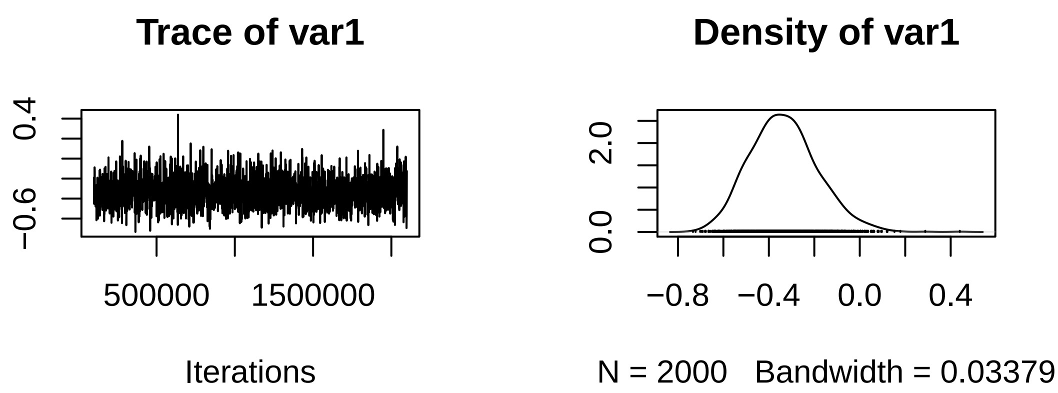
```
##       var1
## -0.4707792
```

```r
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:

```r
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: individuals that flower earlier on average and are more responsive to temperature have higher fitness.

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 here), 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 the var-covar matrix on the latent scale can simply be treated as selection differentials S. By extension, estimates of selection gradients 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
```

```r
# 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
```

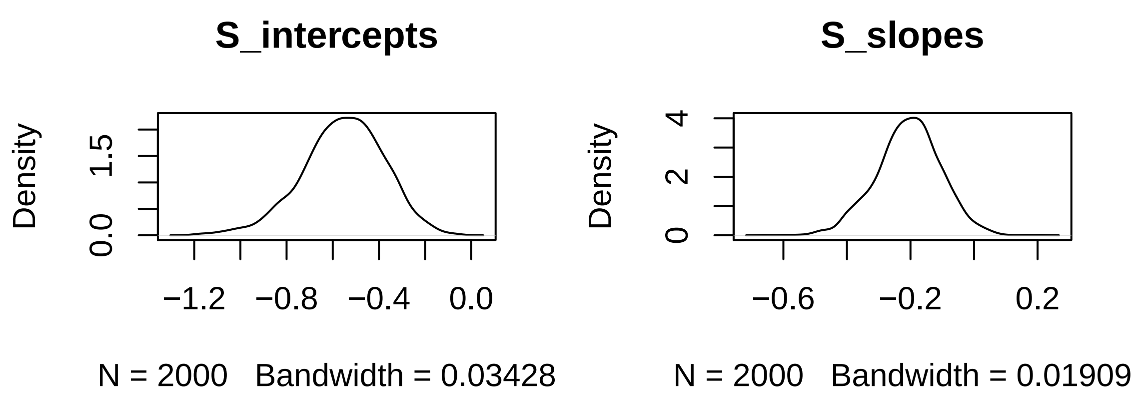
```r
posterior.mode(mcmc(S.modelBV_RR10)) # This is exactly the same as above
```

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

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

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

```r
# 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
```

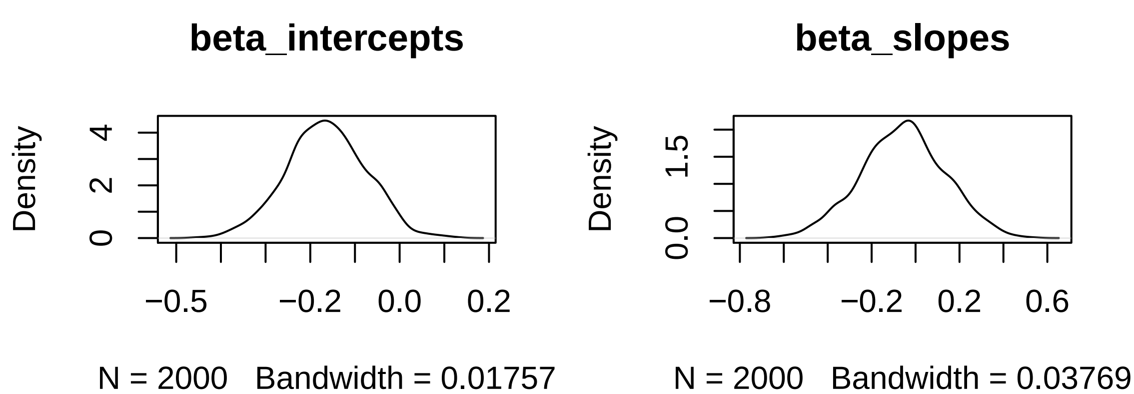
```r
}

# Finally, extract and plot the selection gradients posterior modes
# and 95% credible intervals for both selection on intercepts (trait value)
# and slopes (trait plasticity).

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")
```



The selection differentials are “significant” for both RN intercepts and slopes, and the selection gradient is only significant for the RN intercept (althought the interval almost includes zero). This means that, there is significant total selection (direct + indirect) on intercepts and slopes, but after correcting for the covariance between them, there is direct selection only on the intercept of the RN (i.e. on the average flowering time). Can we say this with the intervals being so close to including zero?

With shoot volume

Stack data:

```r
# Create a single data-set "data.stack14", with single column at start
# to index observations
data.stack14 <- c()
data.stack14$Obs <- 1:(163 + 1162)
```

```r
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 sqrt of mean shoot volume over all years when available, centered
data_5yrs_total<-data_5yrs_total%>%
  mutate(shoot_vol_mean_sqrt=sqrt(shoot_vol_mean),
         cn_shoot_vol_mean_sqrt=scale(shoot_vol_mean_sqrt,center=T,scale=F))
data.stack14$cn_shoot_vol <- c(data_5yrs_total$cn_shoot_vol_mean_sqrt,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)

# 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    59.871875                14 fitness  fitness poisson
## 2   2 new_100 2006    0     5.164343                 4 fitness  fitness poisson
## 3   3 new_101 2006    0    -4.518872                 2 fitness  fitness poisson
## 4   4 new_102 2006    0    18.086848                 6 fitness  fitness poisson
## 5   5 new_103 2006    0     2.068075                 4 fitness  fitness poisson
## 6   6 new_104 2006    0    -6.598538                 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 individuals in fitness
                          # (which is residual variance)
                          us(at.level(variable, "FFD")):Obs,
                          # ^ residual variance within individuals between years
```

```r
                         # (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/User/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	57.429	55.225	59.663	2000	0.000
variablefitness	1.159	1.022	1.289	2000	0.000
at.level(variable, "FFD"):temp	-2.409	-4.062	-0.734	2000	0.009
at.level(variable, "fitness"):cn_shoot_vol	0.021	0.011	0.030	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	25.677	11.423	43.481	2137

```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	3.144	1.820	4.504	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	1.032	0.489	1.632	2000
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.298	-0.648	0.046	2031
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	1.032	0.489	1.632	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	0.800	0.388	1.257	2419
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.084	-0.278	0.097	2000
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.298	-0.648	0.046	2031
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.084	-0.278	0.097	2000
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.379	0.242	0.513	2000
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	19.279	17.519	21.029	2000

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

Table 9: Autocorrelation

	x
variableFFD	0.0224712
variablefitness	0.0043656
at.level(variable, “FFD”):temp	0.0051891
at.level(variable, “fitness”):cn_shoot_vol	-0.0073232

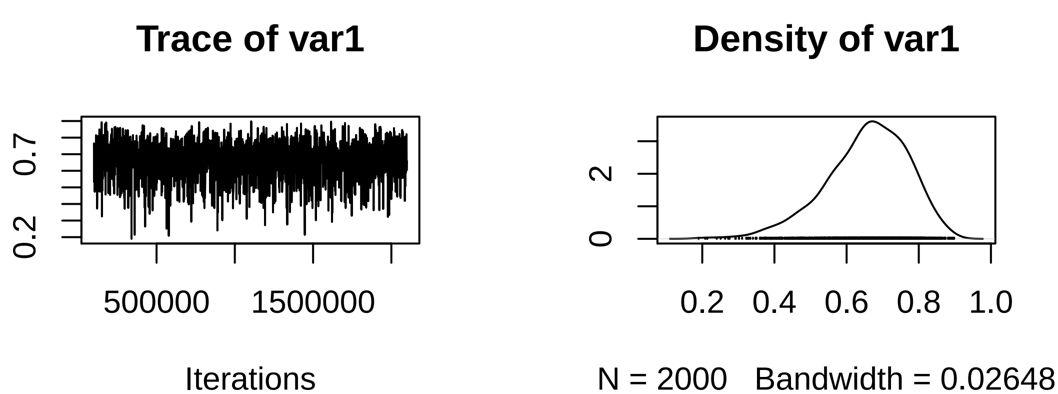
```
kable(diag(autocorr(modelBV_RR14$VCV)[2, , ]),caption="Autocorrelation")
```

Table 10: Autocorrelation

	x
at.level(variable, “FFD”):at.level(variable, “FFD”).year	-0.0332811
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	0.0235869
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	-0.0046305
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	0.0280602
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	-0.0046305
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	-0.0262856
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	0.0040280
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	0.0280602
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	0.0040280
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	0.0102486
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	0.0141450

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)
```

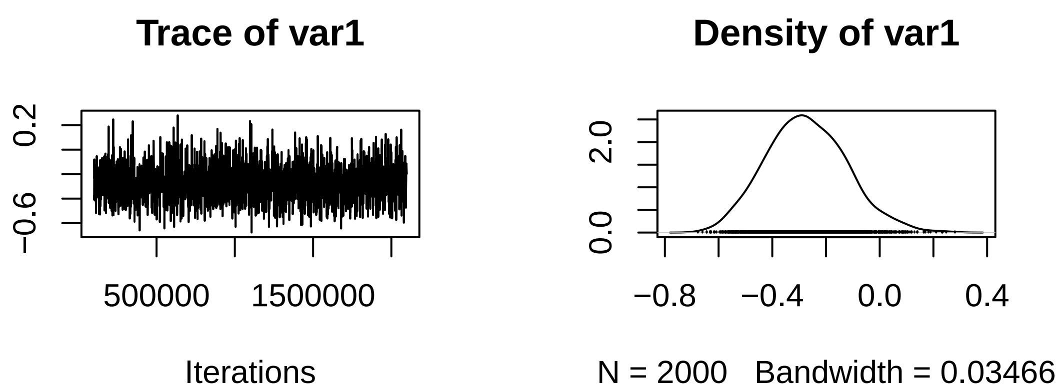
```
##      var1
## 0.6605307
```

```r
HPDinterval(cor_BV_RR_14_intslope)
```

```
##          lower     upper
## var1 0.4278739 0.8641161
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```r
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)
```

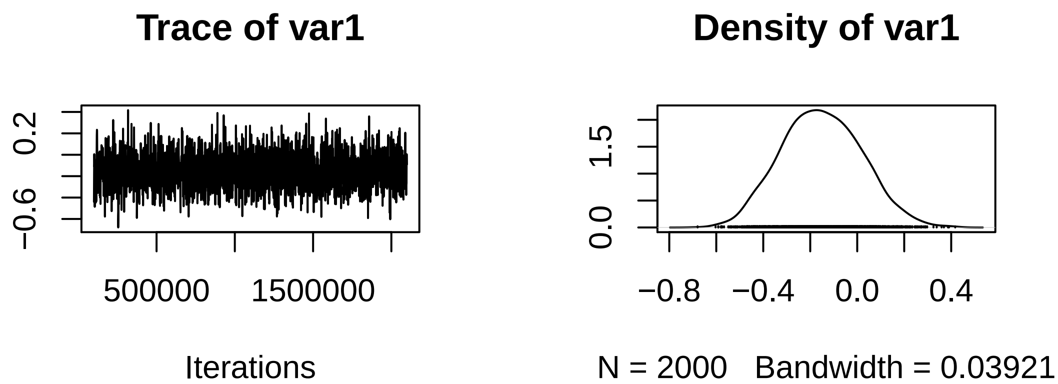
```
##       var1
## -0.2719483
```

```r
HPDinterval(cor_BV_RR_14_intfit)
```

```
##           lower      upper
## var1 -0.5635167 0.01805478
## 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)
```




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

```
##      var1
## -0.156333
```

```r
HPDinterval(cor_BV_RR_14_slopefit)
```

```
##           lower     upper
## var1 -0.4690693 0.1731674
## 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.

```r
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,]  3.1892135  1.06450283 -0.25330514
## [2,]  1.0645028  0.78188070 -0.09338194
## [3,] -0.2533051 -0.09338194  0.35047299
```

```r
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] -0.25330514 -0.09338194
```

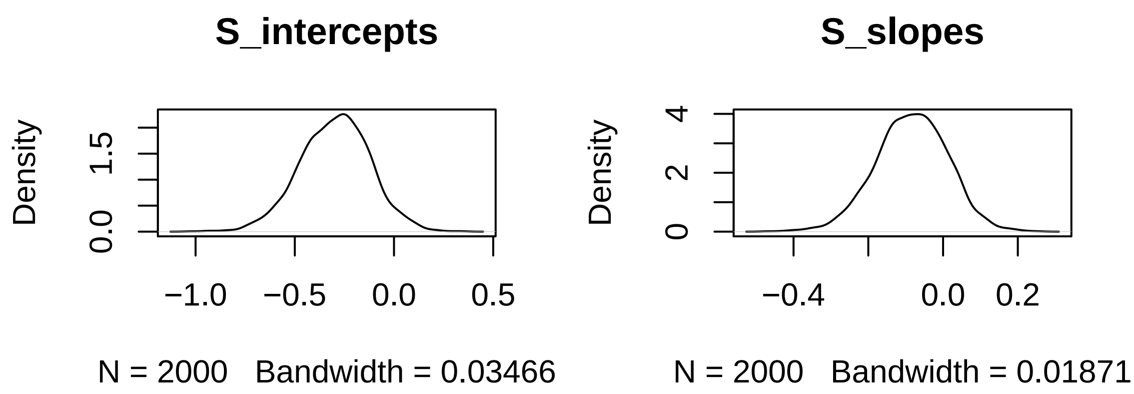
```r
posterior.mode(mcmc(S.modelBV_RR14))
```

```
## S_intercepts     S_slopes
##  -0.25330514  -0.09338194
```

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

```
##                    lower      upper
## S_intercepts -0.6476950 0.04615212
## S_slopes     -0.2782734 0.09686148
## 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 fitness
  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))
```

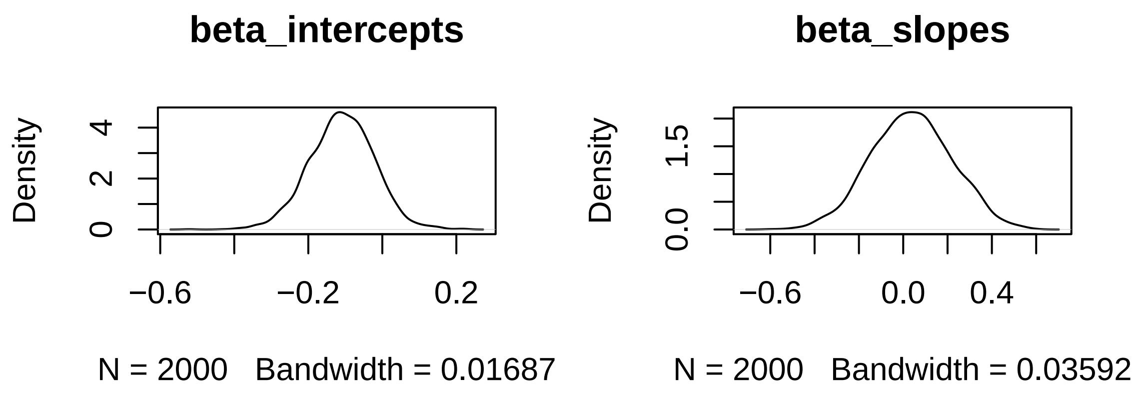
```
## beta_intercepts     beta_slopes
##     -0.06384491     -0.04022447
```

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

```
##                       lower     upper
## beta_intercepts -0.2826591 0.0517351
## beta_slopes     -0.3233725 0.3807692
## attr(,"Probability")
## [1] 0.95
```

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

```r
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 is 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:

```r
# 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)
```

```r
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
```

```r
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/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_RR
```

```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

```r
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

```r
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

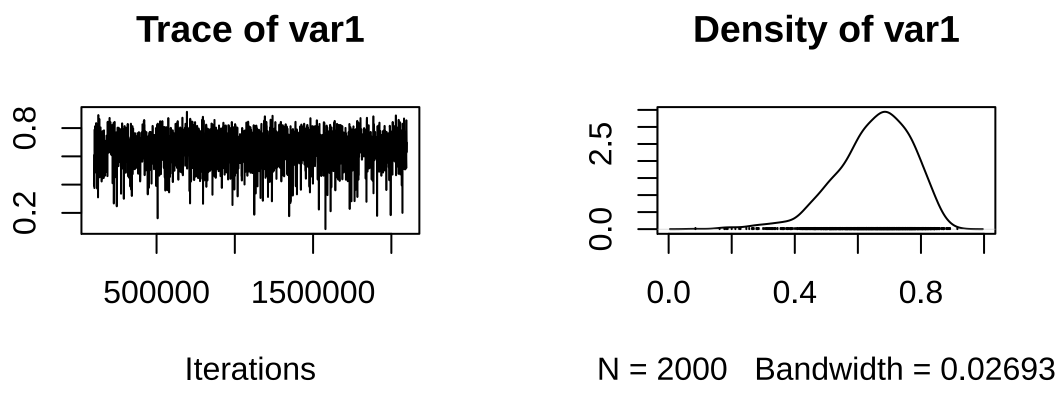
```
kable(diag(autocorr(modelBV_RR12$VCV)[2, , ]),caption="Autocorrelation")
```

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)
```




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

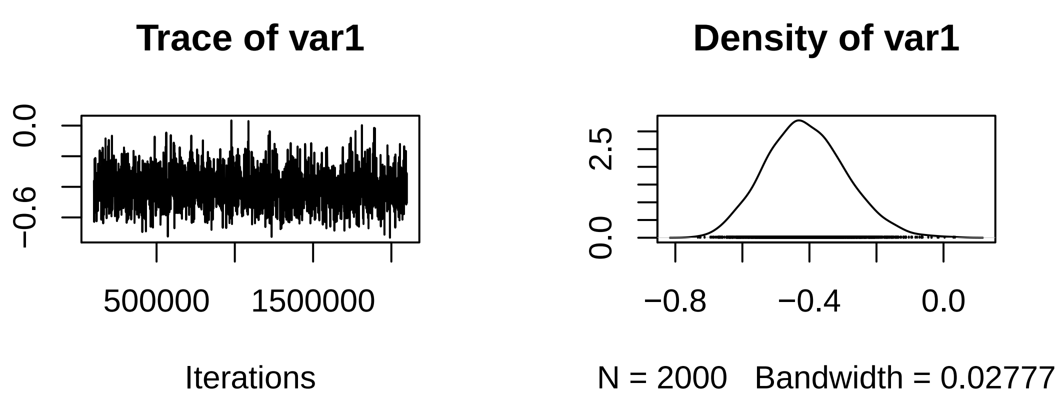
```
##      var1
## 0.6943047
```

```r
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:

```r
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)
```

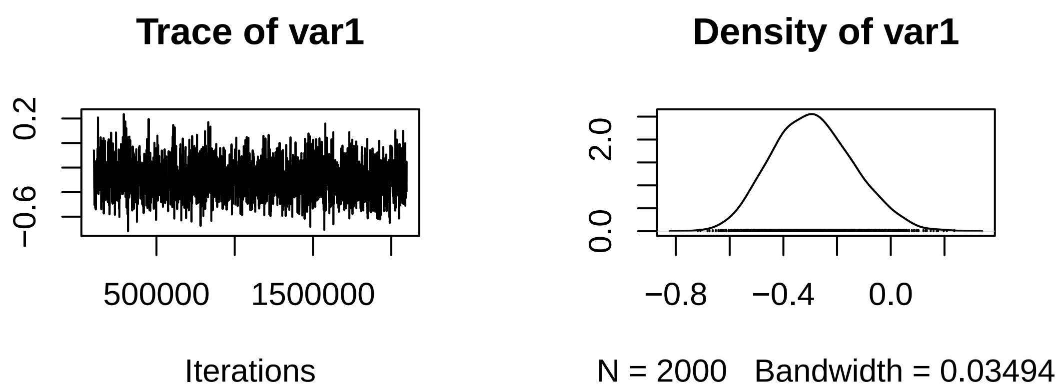
```
##       var1
## -0.4234863
```

```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: individuals that flower earlier on average and are more responsive to temperature have higher fitness.

```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
```

```
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
```

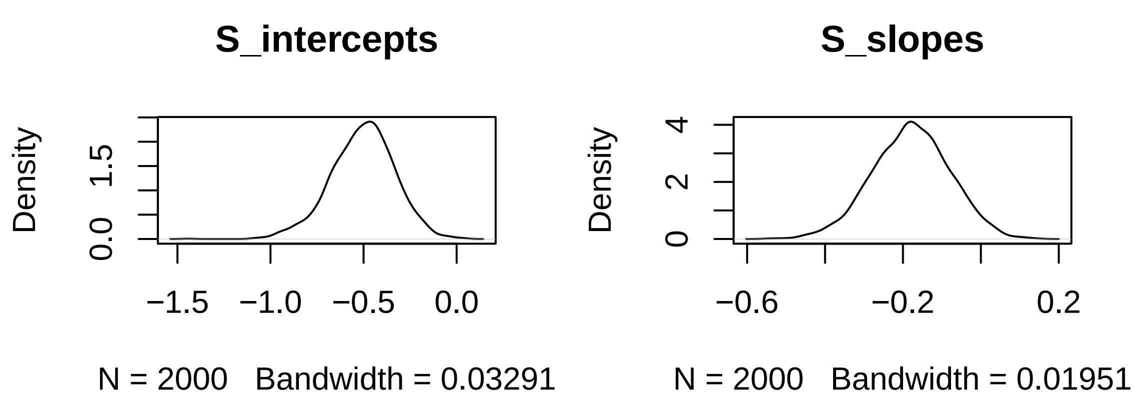
```
posterior.mode(mcmc(S.modelBV_RR12))
```

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

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

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

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




```
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 fitness
  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")
```

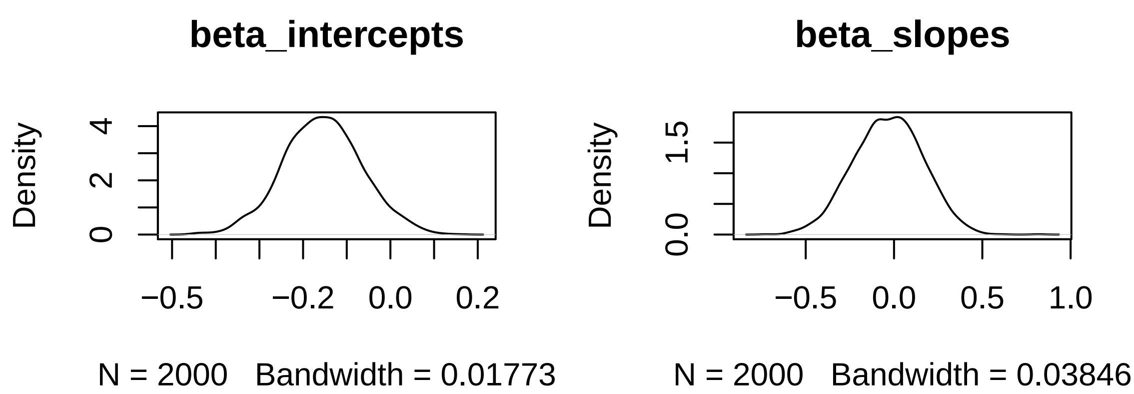
```r
posterior.mode(mcmc(beta_post_RR12))
```

```
## 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:

```r
# 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)
```

```r
# Shoot volume column is only relevant for fitness, but is set to 0 for FFD values
# Using sqrt of mean shoot volume over all years when available, centered
data.stack15$cn_shoot_vol <- c(data_5yrs_total$cn_shoot_vol_mean_sqrt,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    59.871875                16 fitness  fitness poisson
## 2   2 new_100 2006    0     5.164343                 6 fitness  fitness poisson
## 3   3 new_101 2006    0    -4.518872                 3 fitness  fitness poisson
## 4   4 new_102 2006    0    18.086848                 7 fitness  fitness poisson
## 5   5 new_103 2006    0     2.068075                 4 fitness  fitness poisson
## 6   6 new_104 2006    0    -6.598538                 3 fitness  fitness poisson
```

```r
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 indivduals 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/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/modelBV_RR
```

```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	57.389	55.222	59.541	2000	0.000
variablefitness	1.511	1.386	1.631	2000	0.000
at.level(variable, "FFD"):temp	-2.408	-4.030	-0.901	2000	0.005
at.level(variable, "fitness"):cn_shoot_vol	0.015	0.005	0.025	2000	0.006

```
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	25.268	11.089	42.228	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	3.157	1.857	4.456	2111
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	1.025	0.492	1.626	2000
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.328	-0.693	0.014	2000
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	1.025	0.492	1.626	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	0.801	0.419	1.311	2000
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.087	-0.288	0.119	2000
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.328	-0.693	0.014	2000
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.087	-0.288	0.119	2000
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.437	0.296	0.586	1861
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	19.247	17.559	21.035	2209

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

Table 19: Autocorrelation

	x
variableFFD	0.0064935
variablefitness	0.0047522
at.level(variable, "FFD"):temp	0.0253588
at.level(variable, "fitness"):cn_shoot_vol	-0.0311517

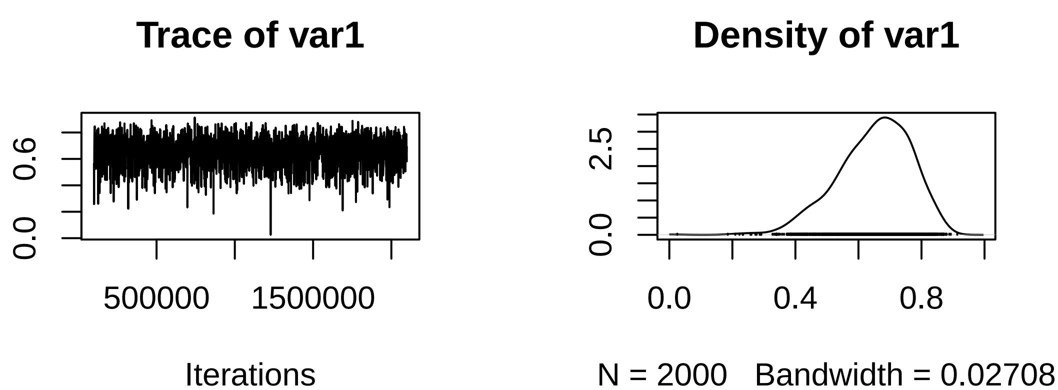
```
kable(diag(autocorr(modelBV_RR15$VCV)[2, , ]),caption="Autocorrelation")
```

Table 20: Autocorrelation

	x
at.level(variable, "FFD"):at.level(variable, "FFD").year	-0.0264170
at.level(variable, "FFD").id:at.level(variable, "FFD").id	-0.0008781
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	-0.0031774
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.0289554
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	-0.0031774
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	-0.0175662
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.0185420
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.0289554
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.0185420
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.0356397
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	-0.0498087

Among-individual correlation between intercepts and slopes for FFD:

```
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)
```




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

```
##      var1
## 0.6696553
```

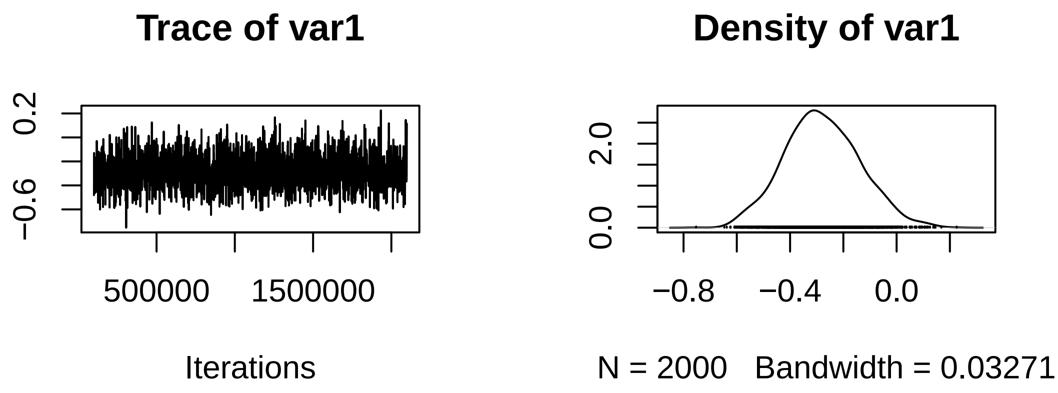
```
HPDinterval(cor_BV_RR_15_intslope)
```

```
##          lower     upper
## var1 0.4145385 0.8476343
## attr(,"Probability")
## [1] 0.95
```

Among-individual correlation between FFD and fitness:

```
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"])*
```

```r
    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)
```

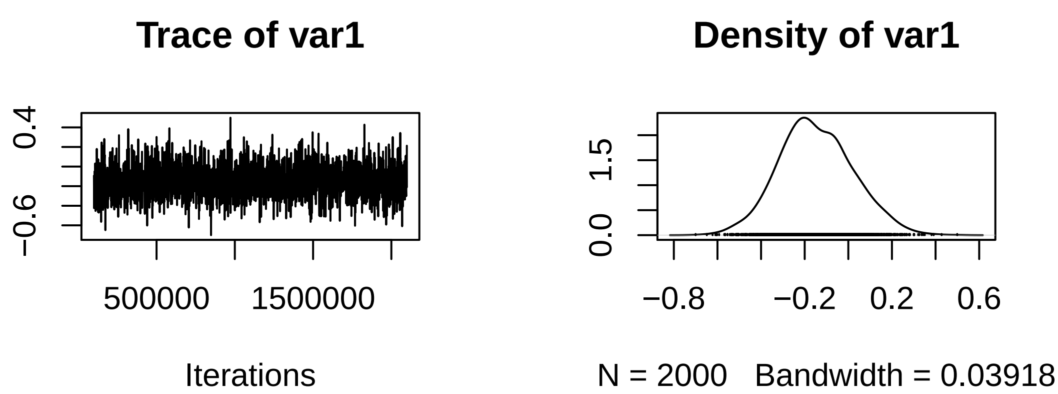
```
##       var1
## -0.3016067
```

```r
HPDinterval(cor_BV_RR_15_intfit)
```

```
##           lower       upper
## var1 -0.5630849 -0.01082592
## attr(,"Probability")
## [1] 0.95
```

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

```r
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)
```




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

```
##       var1
## -0.1756992
```

```r
HPDinterval(cor_BV_RR_15_slopefit)
```

```
##           lower     upper
## var1 -0.4617697 0.1985005
## 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 correlated with the intercept but not with the slope of the RN: Individuals that flower earlier on average have higher fitness but there is not selection on the slope.

```r
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,]  2.8220702  1.0313765 -0.3543252
## [2,]  1.0313765  0.7328403 -0.0880671
## [3,] -0.3543252 -0.0880671  0.4044054
```

```r
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] -0.3543252 -0.0880671
```

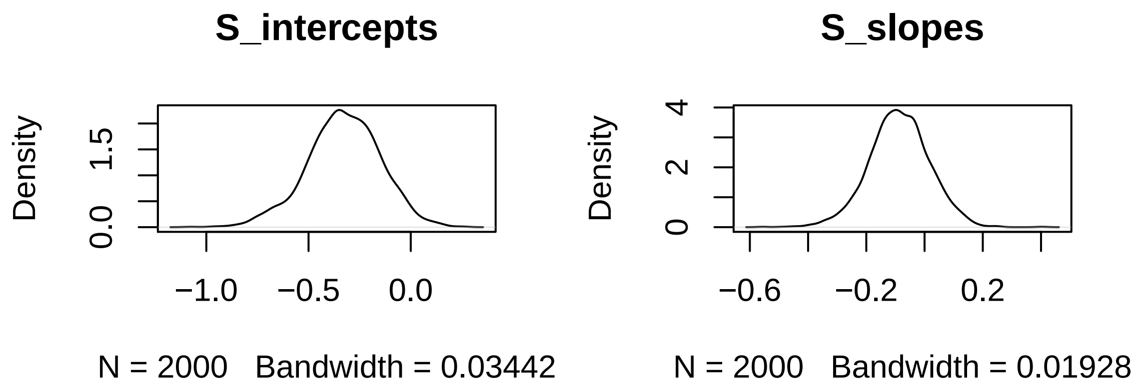
```r
posterior.mode(mcmc(S.modelBV_RR15))
```

```
## S_intercepts     S_slopes
##   -0.3543252   -0.0880671
```

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

```
##                   lower      upper
## S_intercepts -0.6928144 0.01400391
## S_slopes     -0.2884289 0.11850618
## attr(,"Probability")
## [1] 0.95
```

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



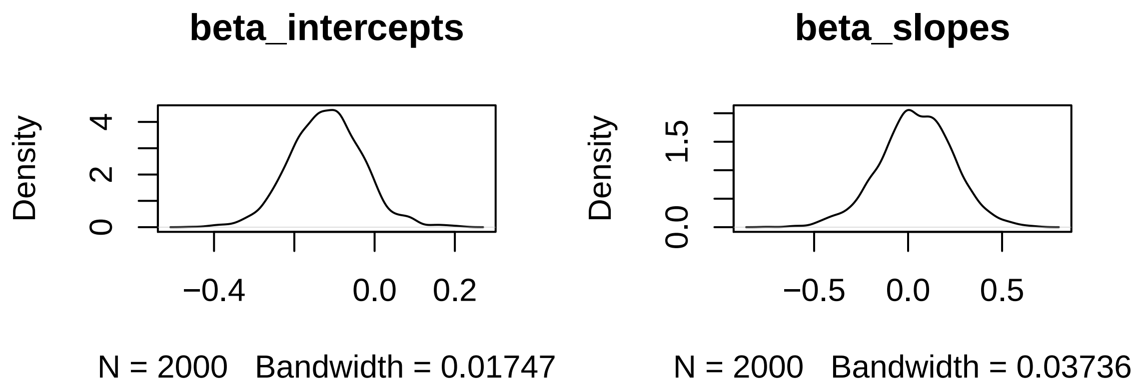

```r
n <- length(modelBV_RR15$VCV[,2])   # sample size
beta_post_RR15 <- matrix(NA, n ,2)

for (i in 1:n) {
  P3 <- matrix(rep(NA, 9), nrow = 3)
  # 3x3 matrix of var-cov for individual X.int, X.slope and fitness
  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
##   -0.0809768393    0.0002624171
```

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

```
##                      lower      upper
## beta_intercepts -0.3027072 0.05207281
## beta_slopes     -0.3436765 0.44075062
## attr(,"Probability")
## [1] 0.95
```

```r
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 is 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!).

```
BLUPs_MCMC<-BLUPs_MCMC%>%
  right_join(shoot_vol_means)

with(BLUPs_MCMC,cor.test(shoot_vol_mean,intercept))  # -0.3685886*
```

```
##
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and intercept
## t = -5.0311, df = 161, p-value = 1.291e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4943021 -0.2277737
## sample estimates:
##        cor
## -0.3685886
```

```
summary(lm(intercept~shoot_vol_mean,BLUPs_MCMC))
```

```
##
## Call:
## lm(formula = intercept ~ shoot_vol_mean, data = BLUPs_MCMC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3271 -0.7281 -0.0096  0.6884  3.6172
##
```

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.855e-01  1.505e-01   3.890 0.000146 ***
## shoot_vol_mean -3.453e-04  6.864e-05  -5.031 1.29e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.21 on 161 degrees of freedom
## Multiple R-squared:  0.1359, Adjusted R-squared:  0.1305
## F-statistic: 25.31 on 1 and 161 DF,  p-value: 1.291e-06
```

```
summary(lm(intercept~log(shoot_vol_mean),BLUPs_MCMC))
```

```
##
## Call:
## lm(formula = intercept ~ log(shoot_vol_mean), data = BLUPs_MCMC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4755 -0.7617 -0.0427  0.7065  3.7756
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.7077     1.1390   5.011 1.41e-06 ***
## log(shoot_vol_mean)  -0.7893     0.1569  -5.031 1.29e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.21 on 161 degrees of freedom
## Multiple R-squared:  0.1359, Adjusted R-squared:  0.1305
## F-statistic: 25.31 on 1 and 161 DF,  p-value: 1.291e-06
```

```
summary(lm(intercept~sqrt(shoot_vol_mean),BLUPs_MCMC))
```

```
##
## Call:
## lm(formula = intercept ~ sqrt(shoot_vol_mean), data = BLUPs_MCMC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4268 -0.7725 -0.0270  0.6648  3.6947
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.441802   0.294117   4.902 2.30e-06 ***
## sqrt(shoot_vol_mean) -0.036954   0.007126  -5.186 6.39e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.205 on 161 degrees of freedom
## Multiple R-squared:  0.1431, Adjusted R-squared:  0.1378
## F-statistic: 26.89 on 1 and 161 DF,  p-value: 6.386e-07
```

```
with(BLUPs_MCMC,cor.test(shoot_vol_mean,slope))  # -0.3762481*
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  shoot_vol_mean and slope
## t = -5.1527, df = 161, p-value = 7.431e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5009926 -0.2361880
## sample estimates:
##        cor
## -0.3762481
```

```
summary(lm(slope~shoot_vol_mean,BLUPs_MCMC))
```

```
##
## Call:
## lm(formula = slope ~ shoot_vol_mean, data = BLUPs_MCMC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47087 -0.30913 -0.02389  0.23690  1.50891
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.434e-01  6.132e-02   3.969 0.000109 ***
## shoot_vol_mean -1.441e-04  2.797e-05  -5.153 7.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4929 on 161 degrees of freedom
## Multiple R-squared:  0.1416, Adjusted R-squared:  0.1362
## F-statistic: 26.55 on 1 and 161 DF,  p-value: 7.431e-07
```

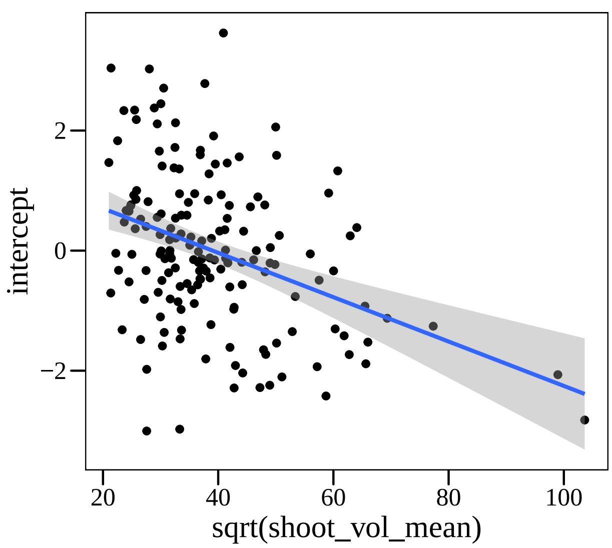
```
summary(lm(slope~log(shoot_vol_mean),BLUPs_MCMC))
```

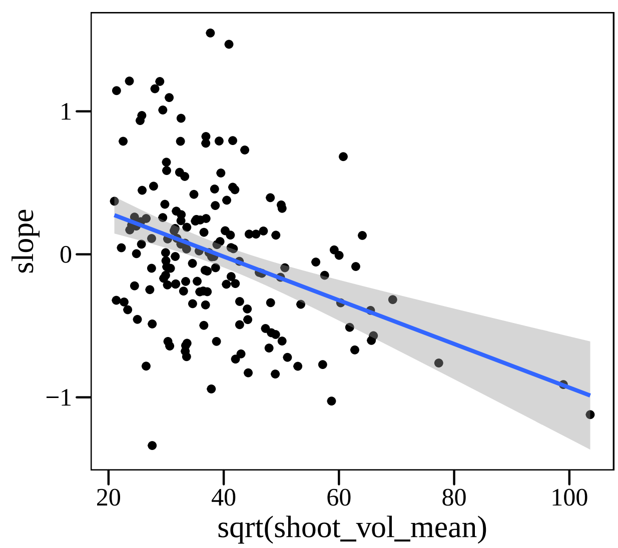
```
##
## Call:
## lm(formula = slope ~ log(shoot_vol_mean), data = BLUPs_MCMC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.52775 -0.31022 -0.02315  0.25478  1.55724
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.31987    0.46603   4.978 1.64e-06 ***
## log(shoot_vol_mean) -0.32093    0.06419  -5.000 1.49e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.495 on 161 degrees of freedom
## Multiple R-squared:  0.1344, Adjusted R-squared:  0.129
## F-statistic:    25 on 1 and 161 DF,  p-value: 1.486e-06
```

```
summary(lm(slope~sqrt(shoot_vol_mean),BLUPs_MCMC))
```

```
##
```

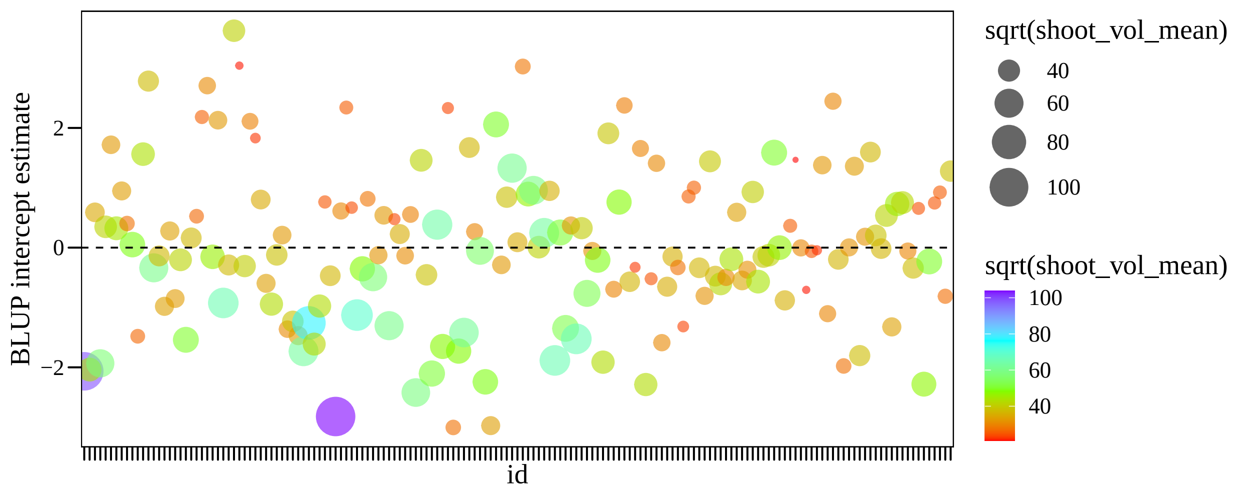
```
## Call:
## lm(formula = slope ~ sqrt(shoot_vol_mean), data = BLUPs_MCMC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51073 -0.30751 -0.01875  0.24499  1.52825
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.594842   0.120022   4.956 1.81e-06 ***
## sqrt(shoot_vol_mean) -0.015270   0.002908  -5.251 4.72e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4916 on 161 degrees of freedom
## Multiple R-squared:  0.1462, Adjusted R-squared:  0.1409
## F-statistic: 27.57 on 1 and 161 DF,  p-value: 4.722e-07
```

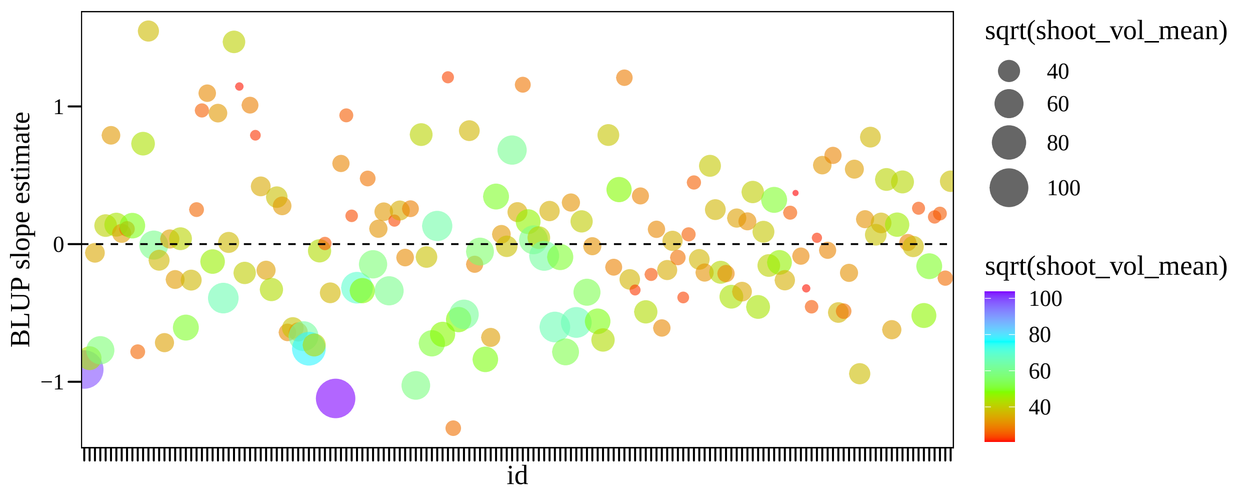


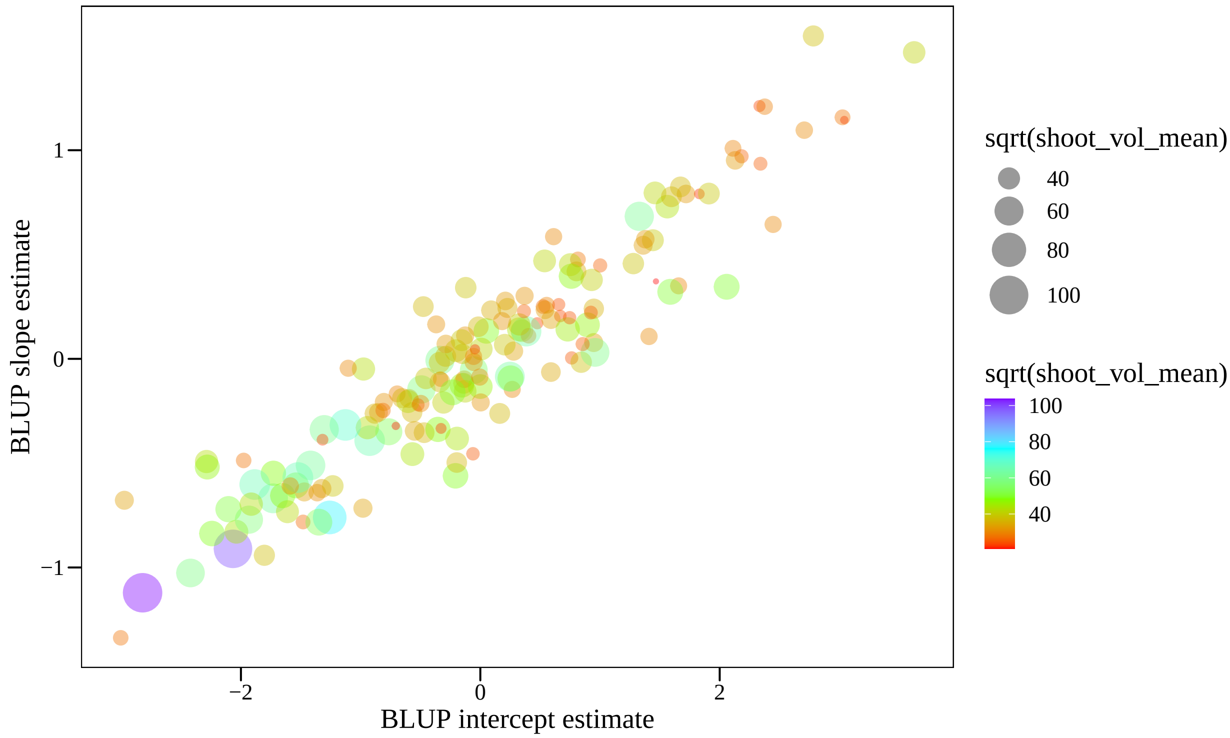


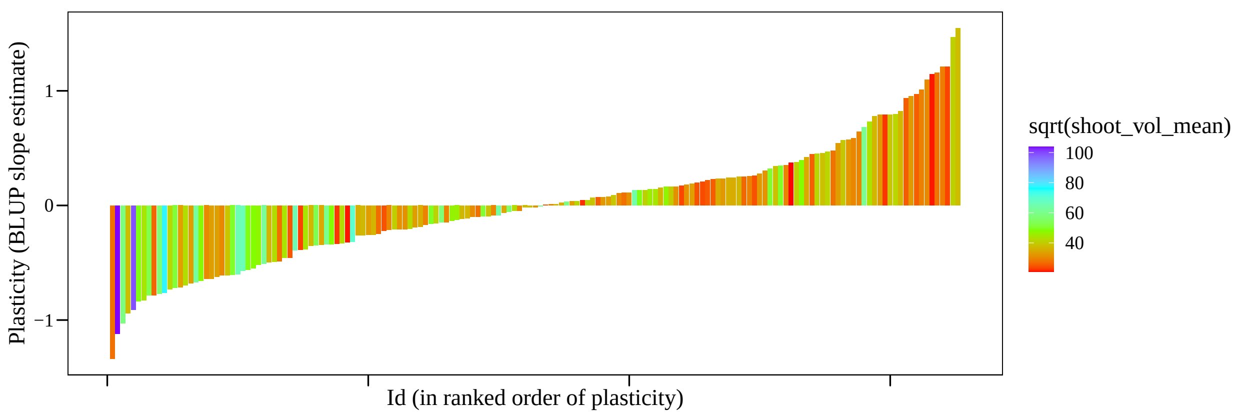
There is a significant negative correlation among size and RN elevation and slope, meaning that larger plants have lower elevations (i.e. flower earlier on average) and slopes (i.e. are more responsive to temperature).

Some graphs for BLUPs showing the size of individuals:


BLUP intercept estimate
2
0
−2
id
sqrt(shoot_vol_mean)
40
60
80
100
sqrt(shoot_vol_mean)
100
80
60
40

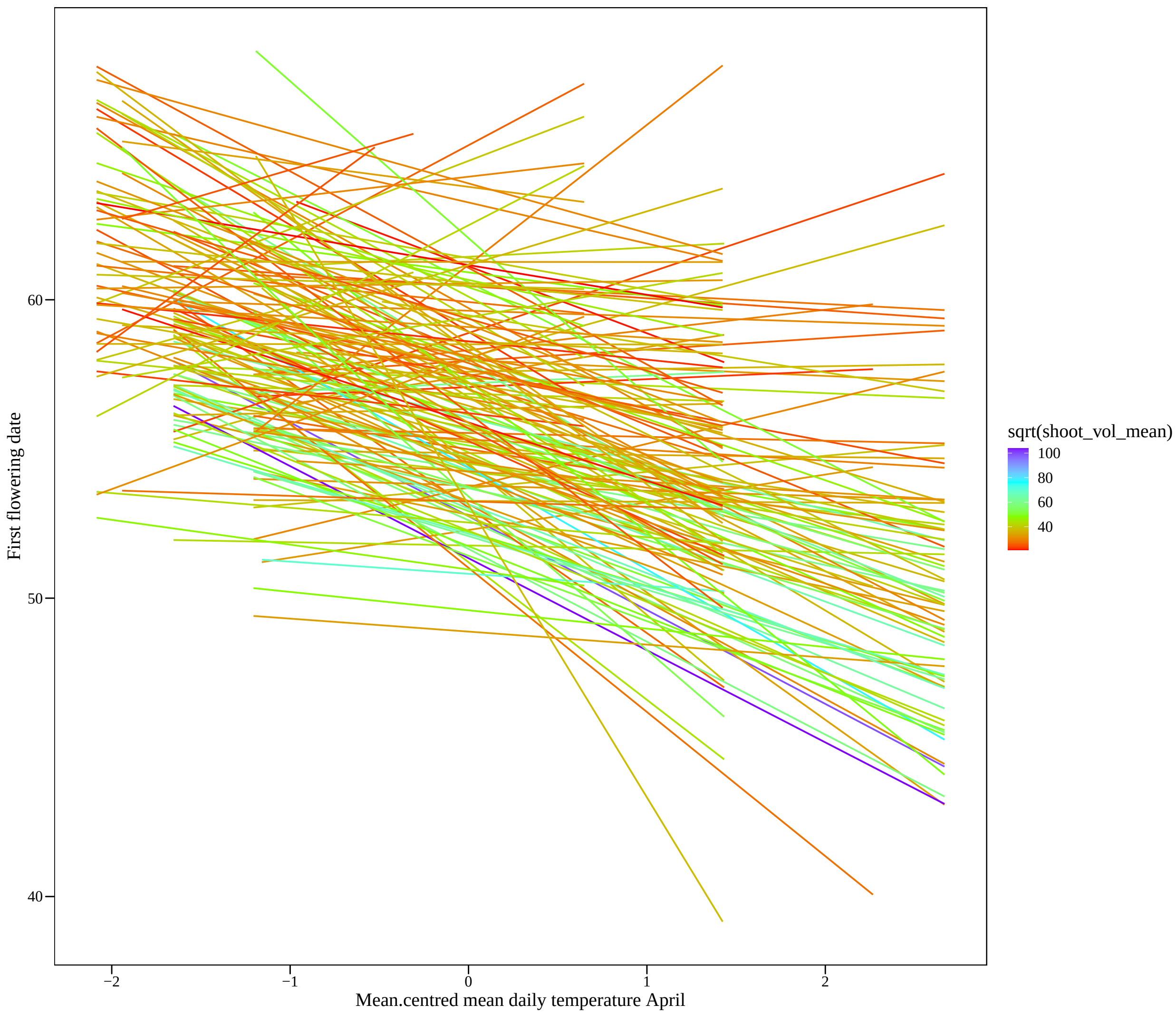

BLUP slope estimate
1
0
−1
id
sqrt(shoot_vol_mean)
40
60
80
100
sqrt(shoot_vol_mean)
100
80
60
40





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.

Plot of all the RNs coloured by size:



Variation in selection among years with BLUPs

Add BLUPs to data set

```
data_5yrs<-data_5yrs%>%left_join(BLUPs_MCMC[1:3])
```

Model with absolute fitness

glmmTMB

```
model_years_abs<-glmmTMB(round(n_intact_seeds)~as.factor(year)*(intercept+slope)+(1|id),
                         data_5yrs,family=poisson) # Model is overdispersed
summary(model_years_abs)
```

Year*(intercept+slope)

```
##  Family: poisson  ( log )
```

```
## Formula:
## round(n_intact_seeds) ~ as.factor(year) * (intercept + slope) +      (1 | id)
## Data: data_5yrs
##
##      AIC      BIC   logLik deviance df.resid
##   9330.9   9669.8  -4598.4   9196.9     1095
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.7109   0.8431
## Number of obs: 1162, groups:  id, 163
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   2.437995   0.116557  20.917  < 2e-16 ***
## as.factor(year)1988          -0.980746   0.079641 -12.315  < 2e-16 ***
## as.factor(year)1989          -0.075326   0.061496  -1.225 0.220612
## as.factor(year)1990          -1.276757   0.093415 -13.668  < 2e-16 ***
## as.factor(year)1991          -0.492739   0.064305  -7.663 1.82e-14 ***
## as.factor(year)1992          -2.263642   0.140060 -16.162  < 2e-16 ***
## as.factor(year)1993          -1.576012   0.094894 -16.608  < 2e-16 ***
## as.factor(year)1994          -2.216969   0.123339 -17.975  < 2e-16 ***
## as.factor(year)1995          -1.562326   0.175701  -8.892  < 2e-16 ***
## as.factor(year)1996          -0.473979   0.113321  -4.183 2.88e-05 ***
## as.factor(year)2006          -0.525100   0.153624  -3.418 0.000631 ***
## as.factor(year)2007          -1.016465   0.153058  -6.641 3.12e-11 ***
## as.factor(year)2008           0.032583   0.148494   0.219 0.826319
## as.factor(year)2009          -2.703636   0.198076 -13.649  < 2e-16 ***
## as.factor(year)2010          -1.856763   0.163790 -11.336  < 2e-16 ***
## as.factor(year)2011          -3.009953   0.202411 -14.871  < 2e-16 ***
## as.factor(year)2012          -1.642082   0.159722 -10.281  < 2e-16 ***
## as.factor(year)2013          -3.671951   0.254337 -14.437  < 2e-16 ***
## as.factor(year)2014          -1.844919   0.173316 -10.645  < 2e-16 ***
## as.factor(year)2015          -0.520190   0.169236  -3.074 0.002114 **
## as.factor(year)2016          -0.395761   0.151479  -2.613 0.008985 **
## as.factor(year)2017          -6.892614   0.862663  -7.990 1.35e-15 ***
## intercept                     0.294961   0.232431   1.269 0.204432
## slope                        -0.487636   0.611191  -0.798 0.424960
## as.factor(year)1988:intercept -0.209776   0.169684  -1.236 0.216355
## as.factor(year)1989:intercept -0.584576   0.134362  -4.351 1.36e-05 ***
## as.factor(year)1990:intercept -0.003847   0.184886  -0.021 0.983400
## as.factor(year)1991:intercept -0.497969   0.140405  -3.547 0.000390 ***
## as.factor(year)1992:intercept -1.103662   0.259493  -4.253 2.11e-05 ***
## as.factor(year)1993:intercept -0.672328   0.223425  -3.009 0.002619 **
## as.factor(year)1994:intercept -0.133129   0.294424  -0.452 0.651149
## as.factor(year)1995:intercept  0.286861   0.427223   0.671 0.501931
## as.factor(year)1996:intercept -0.115980   0.252762  -0.459 0.646341
## as.factor(year)2006:intercept -1.241807   0.323915  -3.834 0.000126 ***
## as.factor(year)2007:intercept -0.472320   0.324321  -1.456 0.145300
## as.factor(year)2008:intercept -0.380650   0.315153  -1.208 0.227115
## as.factor(year)2009:intercept -0.301443   0.435330  -0.692 0.488657
## as.factor(year)2010:intercept -0.704611   0.349438  -2.016 0.043757 *
```

```
## as.factor(year)2011:intercept -0.365427   0.397279  -0.920 0.357665
## as.factor(year)2012:intercept -1.568255   0.335455  -4.675 2.94e-06 ***
## as.factor(year)2013:intercept -0.802623   0.539921  -1.487 0.137132
## as.factor(year)2014:intercept -2.095106   0.346300  -6.050 1.45e-09 ***
## as.factor(year)2015:intercept  0.205301   0.349779   0.587 0.557240
## as.factor(year)2016:intercept -0.809400   0.320670  -2.524 0.011600 *
## as.factor(year)2017:intercept -1.544871   0.746821  -2.069 0.038584 *
## as.factor(year)1988:slope      0.106929   0.429934   0.249 0.803585
## as.factor(year)1989:slope      0.692033   0.317426   2.180 0.029247 *
## as.factor(year)1990:slope     -0.770574   0.449469  -1.714 0.086454 .
## as.factor(year)1991:slope      0.684916   0.340221   2.013 0.044099 *
## as.factor(year)1992:slope      3.110773   0.709274   4.386 1.16e-05 ***
## as.factor(year)1993:slope      1.422437   0.564876   2.518 0.011798 *
## as.factor(year)1994:slope     -1.201084   0.779771  -1.540 0.123486
## as.factor(year)1995:slope      1.907518   0.939811   2.030 0.042389 *
## as.factor(year)1996:slope     -0.433741   0.514451  -0.843 0.399165
## as.factor(year)2006:slope      2.250347   0.811639   2.773 0.005561 **
## as.factor(year)2007:slope      0.148055   0.814821   0.182 0.855816
## as.factor(year)2008:slope     -0.169425   0.792203  -0.214 0.830651
## as.factor(year)2009:slope     -0.411963   1.062227  -0.388 0.698142
## as.factor(year)2010:slope      1.031995   0.865088   1.193 0.232895
## as.factor(year)2011:slope     -0.337062   0.988236  -0.341 0.733048
## as.factor(year)2012:slope      2.817765   0.834236   3.378 0.000731 ***
## as.factor(year)2013:slope      1.838352   1.230209   1.494 0.135086
## as.factor(year)2014:slope      4.011357   0.867944   4.622 3.81e-06 ***
## as.factor(year)2015:slope     -0.544544   0.856842  -0.636 0.525086
## as.factor(year)2016:slope      1.385247   0.804615   1.722 0.085137 .
## as.factor(year)2017:slope     -1.033061   1.788294  -0.578 0.563480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(model_years_abs)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
##                              Chisq Df Pr(>Chisq)
## as.factor(year)          3379.7531 21     <2e-16 ***
## intercept                   2.4932  1     0.1143
## slope                       0.1909  1     0.6621
## as.factor(year):intercept  218.6208 21     <2e-16 ***
## as.factor(year):slope      230.4526 21     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Year*slope Correlation intercept-slope is 0.94, so maybe not good to used them both in the same bebsamodel

```
model_years_abs_slope<-glmmTMB(round(n_intact_seeds)~as.factor(year)*slope+(1|id),
                   data_5yrs,family=poisson) # Model is overdispersed
summary(model_years_abs_slope)
```

```
##  Family: poisson  ( log )
## Formula:          round(n_intact_seeds) ~ as.factor(year) * slope + (1 | id)
## Data: data_5yrs
```

```
##
##      AIC      BIC   logLik deviance df.resid
##   9528.9   9756.5  -4719.5   9438.9     1117
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.7087   0.8418
## Number of obs: 1162, groups:  id, 163
##
## Conditional model:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.48533    0.11546  21.526  < 2e-16 ***
## as.factor(year)1988      -1.02673    0.07762 -13.227  < 2e-16 ***
## as.factor(year)1989      -0.12657    0.05989  -2.113 0.034574 *
## as.factor(year)1990      -1.27617    0.08874 -14.381  < 2e-16 ***
## as.factor(year)1991      -0.53956    0.06280  -8.592  < 2e-16 ***
## as.factor(year)1992      -2.23678    0.13168 -16.986  < 2e-16 ***
## as.factor(year)1993      -1.60794    0.09309 -17.273  < 2e-16 ***
## as.factor(year)1994      -2.26960    0.12247 -18.531  < 2e-16 ***
## as.factor(year)1995      -1.61782    0.17623  -9.180  < 2e-16 ***
## as.factor(year)1996      -0.48254    0.10229  -4.717 2.39e-06 ***
## as.factor(year)2006      -0.51116    0.15190  -3.365 0.000765 ***
## as.factor(year)2007      -1.05157    0.15209  -6.914 4.71e-12 ***
## as.factor(year)2008       0.01221    0.14748   0.083 0.934012
## as.factor(year)2009      -2.73386    0.19697 -13.880  < 2e-16 ***
## as.factor(year)2010      -1.88917    0.16271 -11.610  < 2e-16 ***
## as.factor(year)2011      -3.06292    0.20130 -15.216  < 2e-16 ***
## as.factor(year)2012      -1.55810    0.15608  -9.983  < 2e-16 ***
## as.factor(year)2013      -3.67555    0.24744 -14.855  < 2e-16 ***
## as.factor(year)2014      -1.59158    0.16437  -9.683  < 2e-16 ***
## as.factor(year)2015      -0.84357    0.16413  -5.140 2.75e-07 ***
## as.factor(year)2016      -0.42493    0.14994  -2.834 0.004596 **
## as.factor(year)2017      -6.71930    0.80760  -8.320  < 2e-16 ***
## slope                     0.07369    0.29989   0.246 0.805906
## as.factor(year)1988:slope -0.24248    0.21569  -1.124 0.260937
## as.factor(year)1989:slope -0.51005    0.15259  -3.343 0.000830 ***
## as.factor(year)1990:slope -0.70973    0.20786  -3.414 0.000639 ***
## as.factor(year)1991:slope -0.33824    0.16572  -2.041 0.041244 *
## as.factor(year)1992:slope  0.77191    0.42326   1.824 0.068197 .
## as.factor(year)1993:slope -0.02334    0.23205  -0.101 0.919895
## as.factor(year)1994:slope -1.32546    0.34935  -3.794 0.000148 ***
## as.factor(year)1995:slope  2.34812    0.57057   4.115 3.86e-05 ***
## as.factor(year)1996:slope -0.74239    0.32714  -2.269 0.023248 *
## as.factor(year)2006:slope -0.39684    0.34333  -1.156 0.247743
## as.factor(year)2007:slope -0.78509    0.34220  -2.294 0.021776 *
## as.factor(year)2008:slope -0.93523    0.33663  -2.778 0.005466 **
## as.factor(year)2009:slope -1.00038    0.39682  -2.521 0.011702 *
## as.factor(year)2010:slope -0.43358    0.35411  -1.224 0.220792
## as.factor(year)2011:slope -1.06080    0.40794  -2.600 0.009312 **
## as.factor(year)2012:slope -0.52664    0.34783  -1.514 0.130009
## as.factor(year)2013:slope  0.17356    0.44393   0.391 0.695826
## as.factor(year)2014:slope -0.65448    0.35701  -1.833 0.066772 .
```

```
## as.factor(year)2015:slope -0.07270    0.35343  -0.206 0.837032
## as.factor(year)2016:slope -0.35479    0.33990  -1.044 0.296567
## as.factor(year)2017:slope -4.31250    0.93338  -4.620 3.83e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model_years_abs_slope)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
##                          Chisq Df Pr(>Chisq)
## as.factor(year)       3478.213 21  < 2.2e-16 ***
## slope                   13.349  1  0.0002586 ***
## as.factor(year):slope  183.070 21  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_years_abs_temp<-glmmTMB(round(n_intact_seeds)~cmean_4*(intercept+slope)+(1|id),
                   data_5yrs,family=poisson) # Model is overdispersed
summary(model_years_abs_temp)
```

Temp*(intercept+slope)

```
##  Family: poisson  ( log )
## Formula:          round(n_intact_seeds) ~ cmean_4 * (intercept + slope) + (1 |
##     id)
## Data: data_5yrs
##
##      AIC      BIC   logLik deviance df.resid
##  14844.0  14879.5  -7415.0  14830.0     1155
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.7662   0.8753
## Number of obs: 1162, groups:  id, 163
##
## Conditional model:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.436824   0.071332  20.143  < 2e-16 ***
## cmean_4            0.013259   0.010118   1.310    0.190
## intercept         -0.191939   0.153116  -1.254    0.210
## slope             -0.006874   0.374022  -0.018    0.985
## cmean_4:intercept  0.094507   0.020303   4.655 3.24e-06 ***
## cmean_4:slope     -0.309331   0.048503  -6.378 1.80e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model_years_abs_temp)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
```

```
##                      Chisq Df Pr(>Chisq)
## cmean_4             5.9129  1    0.01503 *
## intercept           1.6890  1    0.19373
## slope               0.0008  1    0.97684
## cmean_4:intercept 21.6673  1  3.243e-06 ***
## cmean_4:slope     40.6728  1  1.800e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_years_abs_temp_slope<-glmmTMB(round(n_intact_seeds)~cmean_4*slope+(1|id),
                   data_5yrs,family=poisson) # Model is overdispersed
summary(model_years_abs_temp_slope)
```

Temp*slope

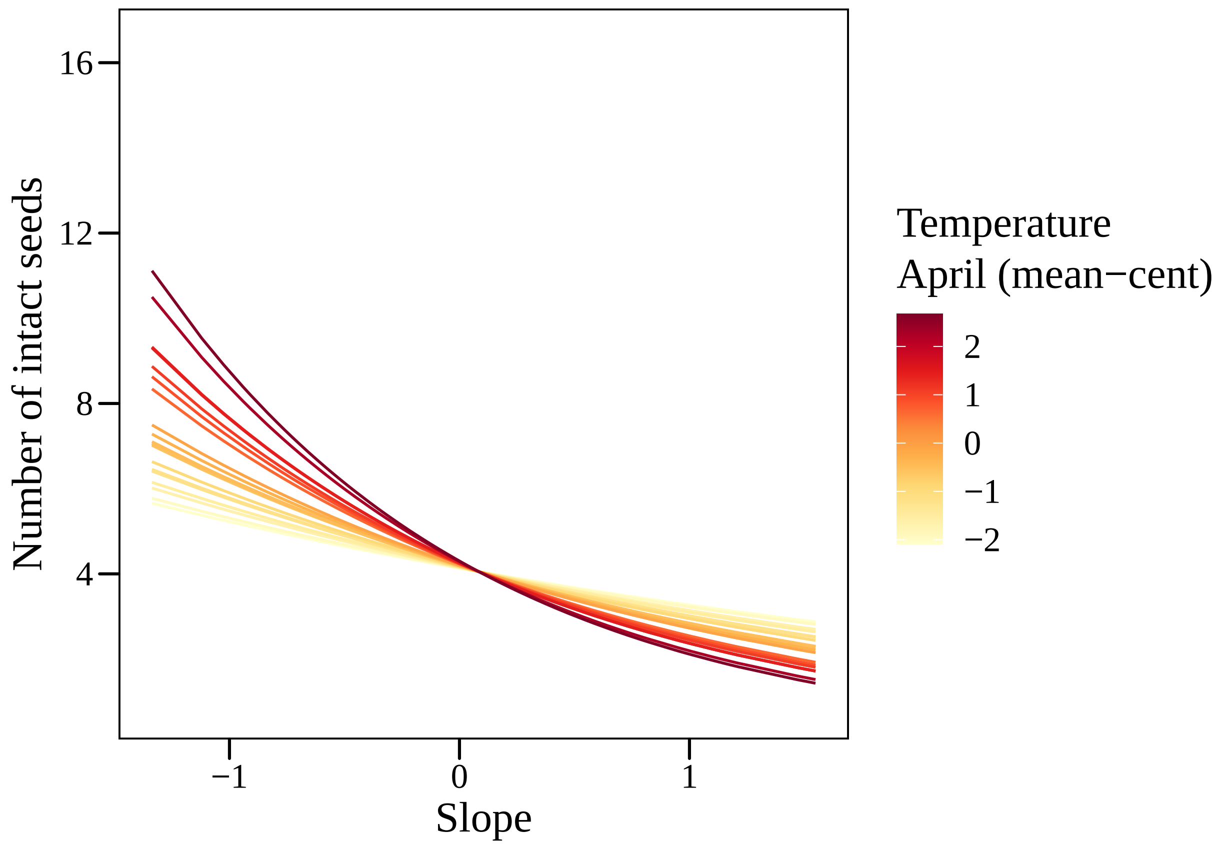
```
##  Family: poisson  ( log )
## Formula:          round(n_intact_seeds) ~ cmean_4 * slope + (1 | id)
## Data: data_5yrs
##
##      AIC      BIC   logLik deviance df.resid
##  14863.5  14888.8  -7426.7  14853.5     1157
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.7682   0.8765
## Number of obs: 1162, groups:  id, 163
##
## Conditional model:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.436613   0.071420  20.115  < 2e-16 ***
## cmean_4        0.008399   0.010062   0.835  0.40390
## slope         -0.443301   0.135574  -3.270  0.00108 **
## cmean_4:slope -0.099973   0.018093  -5.526 3.28e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model_years_abs_temp_slope)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
##                 Chisq Df Pr(>Chisq)
## cmean_4        5.6993  1  0.0169716 *
## slope         11.3943  1  0.0007367 ***
## cmean_4:slope 30.5325  1  3.283e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
myPalette <- colorRampPalette(brewer.pal(11, "YlOrRd"))
ggpredict(model_years_abs_temp_slope,
```

```r
         terms = c("slope [all]", "cmean_4 [all]"))%>%
  ggplot(aes(x,predicted, ymin = conf.low, ymax = conf.high, colour = group, fill = group))+
  geom_line(aes(color=as.numeric(as.character(group)))) +
  scale_colour_gradientn(colours = myPalette(100)) +
  # geom_ribbon(alpha = .1, colour = NA)+
  my_theme()+xlab("Slope")+ylab("Number of intact seeds")+
  theme(legend.position="right")+labs(colour="Temperature\nApril (mean-cent)")
```



Plot

```r
model_years_abs_vol_slope<-glmmTMB(round(n_intact_seeds)~sqrt(shoot_vol)*slope+(1|id),
                   subset(data_5yrs,shoot_vol>0),family=poisson) # Model is overdispersed
summary(model_years_abs_vol_slope)
```

Volume*slope

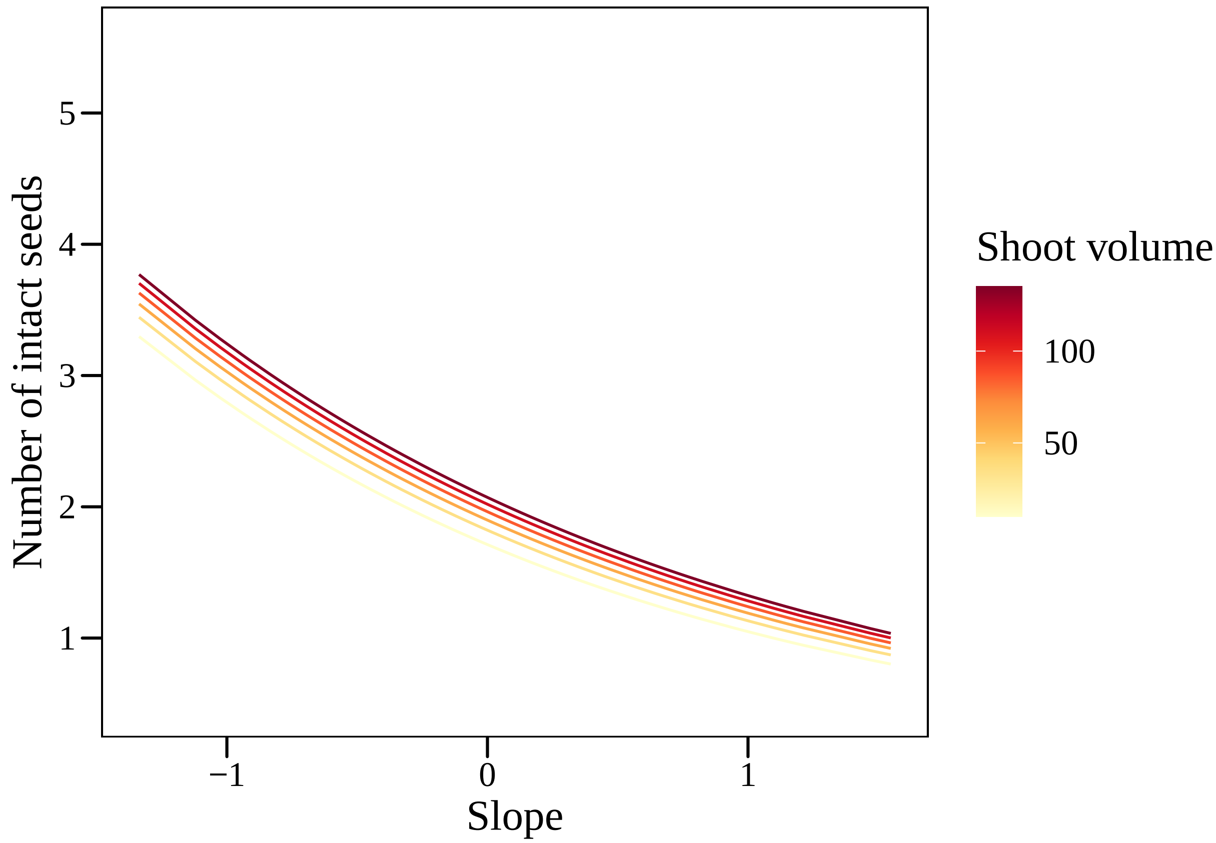
```
##  Family: poisson  ( log )
## Formula:          round(n_intact_seeds) ~ sqrt(shoot_vol) * slope + (1 | id)
## Data: subset(data_5yrs, shoot_vol > 0)
##
##      AIC      BIC   logLik deviance df.resid
##  12854.4  12879.4  -6422.2  12844.4     1087
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
```

```
##  id     (Intercept) 0.7476   0.8647
## Number of obs: 1092, groups:  id, 163
##
## Conditional model:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.4665506  0.0804786   5.797 6.74e-09 ***
## sqrt(shoot_vol)        0.0225282  0.0009313  24.191  < 2e-16 ***
## slope                 -0.5058776  0.1464229  -3.455 0.000550 ***
## sqrt(shoot_vol):slope  0.0050016  0.0013828   3.617 0.000298 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model_years_abs_vol_slope)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
##                          Chisq Df Pr(>Chisq)
## sqrt(shoot_vol)       769.1051  1  < 2.2e-16 ***
## slope                   5.0355  1   0.024834 *
## sqrt(shoot_vol):slope  13.0830  1   0.000298 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
ggpredict(model_years_abs_vol_slope,
          terms = c("slope [all]", "shoot_vol [10:150 by=25]"))%>%
  ggplot(aes(x,predicted, ymin = conf.low, ymax = conf.high, colour = group, fill = group))+
  geom_line(aes(color=as.numeric(as.character(group)))) +
  scale_colour_gradientn(colours = myPalette(100)) +
  # geom_ribbon(alpha = .1, colour = NA)+
  my_theme()+xlab("Slope")+ylab("Number of intact seeds")+
  theme(legend.position="right")+labs(colour="Shoot volume")
```



Plot

```
model_years_abs_temp_slope_vol<-glmmTMB(round(n_intact_seeds)~cmean_4*slope+sqrt(shoot_vol)+(1|id),
                    data_5yrs,family=poisson) # Model is overdispersed
summary(model_years_abs_temp_slope_vol)
```

Temp*slope+volume

```
##  Family: poisson  ( log )
## Formula:
## round(n_intact_seeds) ~ cmean_4 * slope + sqrt(shoot_vol) + (1 |      id)
## Data: data_5yrs
##
##      AIC      BIC   logLik deviance df.resid
##  12871.5  12901.4  -6429.7  12859.5     1087
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.7548   0.8688
## Number of obs: 1093, groups:  id, 163
##
## Conditional model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.5426802  0.0783749   6.924 4.39e-12 ***
## cmean_4             0.0237661  0.0103825   2.289   0.0221 *
```

```
## slope           -0.3030030  0.1359790  -2.228   0.0259 *
## sqrt(shoot_vol)  0.0203699  0.0007549  26.982  < 2e-16 ***
## cmean_4:slope   -0.0152054  0.0187598  -0.811   0.4176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model_years_abs_temp_slope_vol)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
##                    Chisq Df Pr(>Chisq)
## cmean_4           6.7878  1   0.009178 **
## slope             5.0499  1   0.024627 *
## sqrt(shoot_vol) 728.0279  1  < 2.2e-16 ***
## cmean_4:slope     0.6570  1   0.417636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_years_abs_temp_slope_vol_int<-glmmTMB(round(n_intact_seeds)~cmean_4*slope+
                                              sqrt(shoot_vol)*slope+(1|id),
                   data_5yrs,family=poisson) # Model is overdispersed
summary(model_years_abs_temp_slope_vol_int)
```

Temp*slope+volume*slope

```
##  Family: poisson  ( log )
## Formula:
## round(n_intact_seeds) ~ cmean_4 * slope + sqrt(shoot_vol) * slope +
##     (1 | id)
## Data: data_5yrs
##
##      AIC      BIC   logLik deviance df.resid
##  12856.1  12891.1  -6421.0  12842.1     1086
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.7486   0.8652
## Number of obs: 1093, groups:  id, 163
##
## Conditional model:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.4622146  0.0806029   5.734 9.78e-09 ***
## cmean_4               0.0282738  0.0104680   2.701  0.00691 **
## slope                -0.5351949  0.1465446  -3.652  0.00026 ***
## sqrt(shoot_vol)       0.0226656  0.0009347  24.249  < 2e-16 ***
## cmean_4:slope        -0.0137706  0.0188457  -0.731  0.46496
## slope:sqrt(shoot_vol)  0.0057695  0.0013808   4.179 2.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model_years_abs_temp_slope_vol_int)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: round(n_intact_seeds)
##                         Chisq Df Pr(>Chisq)
## cmean_4                9.1632  1   0.002469 **
## slope                  5.0390  1   0.024783 *
## sqrt(shoot_vol)      740.4748  1  < 2.2e-16 ***
## cmean_4:slope          0.5339  1   0.464962
## slope:sqrt(shoot_vol)  17.4601  1  2.934e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MCMCglmm -> USE

For both technical and philosophical reasons, MCMCglmm always adds an observation-level variance (referred to in MCMCglmm as the "R-structure", for "residual structure"), corresponding to an overdispersion term.

MCMCglmm assumes additive overdispersion and this is responsible for the residual term in the model.

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

prior_years <- list(G = list(G1 = list(V = diag(1), nu = 1)), # for random effect of id
                 R = list(R1 = list(V = diag(1), nu = 2)))
```

Year*(intercept+slope) How to test if the interactions year:intercept and year:slope are significant? We should use year as a factor. Fitting model with and wihtout year and the interactions, and looking at difference in DIC?

```r
model_years_abs_bayes <- MCMCglmm(round(n_intact_seeds)~as.factor(year)*(intercept+slope),
                     random = ~ id,
                     rcov = ~units,
                     data = data_5yrs,
                     prior = prior_years,
                     family = "poisson",
                     nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/objects/m
```

```r
summary(model_years_abs_bayes)
```

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3881.338
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
```

```
## id    0.6812   0.3311    1.041     1839
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     3.569    3.032    4.198     2000
##
##  Location effects: round(n_intact_seeds) ~ as.factor(year) * (intercept + slope)
##
##                               post.mean   l-95% CI   u-95% CI eff.samp  pMCMC
## (Intercept)                    1.792638   1.106266   2.405192     2000 <5e-04
## as.factor(year)1988           -1.447324  -2.307688  -0.625641     2000 <5e-04
## as.factor(year)1989            0.008218  -0.807122   0.910275     2000  0.998
## as.factor(year)1990           -2.312054  -3.199565  -1.275649     2038 <5e-04
## as.factor(year)1991           -0.639726  -1.553295   0.142572     2000  0.138
## as.factor(year)1992           -2.899036  -3.880700  -1.816680     2000 <5e-04
## as.factor(year)1993           -2.222116  -3.113460  -1.244067     2000 <5e-04
## as.factor(year)1994           -3.424444  -4.530420  -2.484102     2000 <5e-04
## as.factor(year)1995           -1.775523  -3.165540  -0.350371     1541  0.015
## as.factor(year)1996           -0.588201  -2.009143   0.788543     2401  0.400
## as.factor(year)2006           -0.856821  -1.768988   0.056429     1758  0.082
## as.factor(year)2007           -1.349402  -2.188241  -0.552886     2000 <5e-04
## as.factor(year)2008           -0.044020  -0.844227   0.804314     1903  0.922
## as.factor(year)2009           -3.468162  -4.507577  -2.461341     2000 <5e-04
## as.factor(year)2010           -2.187714  -2.986351  -1.270471     1737 <5e-04
## as.factor(year)2011           -3.705919  -4.587384  -2.689823     2000 <5e-04
## as.factor(year)2012           -2.940902  -3.803714  -2.058559     2000 <5e-04
## as.factor(year)2013           -4.756968  -5.874369  -3.576136     2000 <5e-04
## as.factor(year)2014           -2.312076  -3.363947  -1.351889     1947 <5e-04
## as.factor(year)2015           -1.254514  -2.423955  -0.066486     2000  0.040
## as.factor(year)2016           -0.453856  -1.308824   0.359154     1764  0.285
## as.factor(year)2017           -7.756420 -10.044784  -5.683148     2000 <5e-04
## intercept                      0.505559  -0.815388   1.839843     2000  0.449
## slope                         -1.295669  -4.691039   1.968207     2000  0.447
## as.factor(year)1988:intercept  0.158758  -1.679191   1.914878     2000  0.869
## as.factor(year)1989:intercept -1.113109  -2.874415   0.516877     2000  0.197
## as.factor(year)1990:intercept  0.508924  -1.231355   2.324772     2000  0.579
## as.factor(year)1991:intercept -0.914326  -2.570523   0.850385     2000  0.296
## as.factor(year)1992:intercept -1.977543  -3.972709  -0.146570     2000  0.040
## as.factor(year)1993:intercept -0.692470  -2.481128   1.511514     2000  0.472
## as.factor(year)1994:intercept -0.610111  -2.926602   1.688907     2000  0.632
## as.factor(year)1995:intercept  0.131500  -3.018614   3.150703     2000  0.932
## as.factor(year)1996:intercept  0.668721  -2.141737   3.740528     2093  0.684
## as.factor(year)2006:intercept -2.039022  -4.024355  -0.142508     2000  0.037
## as.factor(year)2007:intercept -0.633773  -2.443485   1.024311     2000  0.487
## as.factor(year)2008:intercept  0.124761  -1.735294   1.878495     2000  0.880
## as.factor(year)2009:intercept -0.368942  -2.868319   1.884229     2184  0.767
## as.factor(year)2010:intercept -1.109202  -2.736771   0.893158     1834  0.231
## as.factor(year)2011:intercept -0.548810  -2.578651   1.373837     2000  0.568
## as.factor(year)2012:intercept -2.651167  -4.660869  -0.851285     2000  0.005
## as.factor(year)2013:intercept -1.963208  -4.771666   0.884054     2556  0.166
## as.factor(year)2014:intercept -2.775993  -4.815969  -0.658743     2000  0.011
## as.factor(year)2015:intercept -0.792721  -3.133214   1.494695     2000  0.513
## as.factor(year)2016:intercept -1.478081  -3.283770   0.275556     2000  0.118
```

```
## as.factor(year)2017:intercept  -0.765733  -4.598541   3.105679     1849  0.689
## as.factor(year)1988:slope       0.368012  -4.435981   5.162901     2000  0.879
## as.factor(year)1989:slope       1.116089  -3.111907   5.592440     2152  0.616
## as.factor(year)1990:slope      -3.203244  -8.156140   1.378665     2000  0.187
## as.factor(year)1991:slope       2.405614  -1.734751   7.001749     2000  0.262
## as.factor(year)1992:slope       7.389746   1.616898  12.907866     2169  0.008
## as.factor(year)1993:slope       1.701185  -3.398504   6.623028     1990  0.514
## as.factor(year)1994:slope      -0.470418  -6.797593   5.741822     2000  0.865
## as.factor(year)1995:slope       2.936876  -3.873240  10.250420     2000  0.410
## as.factor(year)1996:slope      -2.292490  -8.787534   4.095131     2000  0.498
## as.factor(year)2006:slope       3.770353  -0.668066   8.646883     2000  0.105
## as.factor(year)2007:slope       0.088654  -3.961997   4.487431     2000  0.969
## as.factor(year)2008:slope      -0.942407  -5.131993   3.575292     2000  0.674
## as.factor(year)2009:slope      -0.470634  -5.986121   5.409503     1961  0.884
## as.factor(year)2010:slope       2.090794  -2.731583   6.092342     1851  0.359
## as.factor(year)2011:slope      -0.261271  -5.284891   4.541216     2000  0.904
## as.factor(year)2012:slope       4.961654   0.004169   9.171023     2000  0.034
## as.factor(year)2013:slope       5.140267  -1.857211  11.580190     2000  0.114
## as.factor(year)2014:slope       5.705971   0.540604  10.662022     2000  0.026
## as.factor(year)2015:slope       2.483362  -3.107420   7.950871     2000  0.375
## as.factor(year)2016:slope       2.946498  -1.516725   7.283956     2000  0.191
## as.factor(year)2017:slope      -1.339090 -11.370240   8.193672     1856  0.793
##
## (Intercept)                    ***
## as.factor(year)1988            ***
## as.factor(year)1989
## as.factor(year)1990            ***
## as.factor(year)1991
## as.factor(year)1992            ***
## as.factor(year)1993            ***
## as.factor(year)1994            ***
## as.factor(year)1995            *
## as.factor(year)1996
## as.factor(year)2006            .
## as.factor(year)2007            ***
## as.factor(year)2008
## as.factor(year)2009            ***
## as.factor(year)2010            ***
## as.factor(year)2011            ***
## as.factor(year)2012            ***
## as.factor(year)2013            ***
## as.factor(year)2014            ***
## as.factor(year)2015            *
## as.factor(year)2016
## as.factor(year)2017            ***
## intercept
## slope
## as.factor(year)1988:intercept
## as.factor(year)1989:intercept
## as.factor(year)1990:intercept
## as.factor(year)1991:intercept
## as.factor(year)1992:intercept *
## as.factor(year)1993:intercept
## as.factor(year)1994:intercept
```

```
## as.factor(year)1995:intercept
## as.factor(year)1996:intercept
## as.factor(year)2006:intercept *
## as.factor(year)2007:intercept
## as.factor(year)2008:intercept
## as.factor(year)2009:intercept
## as.factor(year)2010:intercept
## as.factor(year)2011:intercept
## as.factor(year)2012:intercept **
## as.factor(year)2013:intercept
## as.factor(year)2014:intercept *
## as.factor(year)2015:intercept
## as.factor(year)2016:intercept
## as.factor(year)2017:intercept
## as.factor(year)1988:slope
## as.factor(year)1989:slope
## as.factor(year)1990:slope
## as.factor(year)1991:slope
## as.factor(year)1992:slope     **
## as.factor(year)1993:slope
## as.factor(year)1994:slope
## as.factor(year)1995:slope
## as.factor(year)1996:slope
## as.factor(year)2006:slope
## as.factor(year)2007:slope
## as.factor(year)2008:slope
## as.factor(year)2009:slope
## as.factor(year)2010:slope
## as.factor(year)2011:slope
## as.factor(year)2012:slope     *
## as.factor(year)2013:slope
## as.factor(year)2014:slope     *
## as.factor(year)2015:slope
## as.factor(year)2016:slope
## as.factor(year)2017:slope
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_years_abs_slope_bayes <- MCMCglmm(round(n_intact_seeds)~as.factor(year)*slope,
                     random = ~ id,
                     rcov = ~units,
                     data = data_5yrs,
                     prior = prior_years,
                     family = "poisson",
                     nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_slope_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/obj
```

```r
summary(model_years_abs_slope_bayes)
```

Year*slope

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3898.378
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id    0.6693   0.3613    1.033     2000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units      3.48    2.935    4.083     2000
##
##  Location effects: round(n_intact_seeds) ~ as.factor(year) * slope
##
##                            post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)                  1.84092  1.17427  2.49464     2000 <5e-04 ***
## as.factor(year)1988         -1.44089 -2.28974 -0.50161     2000 <5e-04 ***
## as.factor(year)1989         -0.04447 -0.85532  0.86108     2000  0.894
## as.factor(year)1990         -2.21308 -3.14612 -1.26263     2000 <5e-04 ***
## as.factor(year)1991         -0.68051 -1.49888  0.15828     2000  0.125
## as.factor(year)1992         -2.82051 -3.75473 -1.80120     2000 <5e-04 ***
## as.factor(year)1993         -2.26499 -3.21540 -1.35006     2154 <5e-04 ***
## as.factor(year)1994         -3.44553 -4.43487 -2.45578     2000 <5e-04 ***
## as.factor(year)1995         -1.84170 -3.14961 -0.41264     1918  0.005 **
## as.factor(year)1996         -0.41062 -1.57937  0.77088     2253  0.493
## as.factor(year)2006         -0.78123 -1.64762  0.07251     2000  0.072 .
## as.factor(year)2007         -1.37788 -2.24892 -0.61692     2000 <5e-04 ***
## as.factor(year)2008         -0.05932 -0.84921  0.79129     2000  0.905
## as.factor(year)2009         -3.46243 -4.49356 -2.49093     2000 <5e-04 ***
## as.factor(year)2010         -2.20175 -3.03836 -1.26773     2000 <5e-04 ***
## as.factor(year)2011         -3.70838 -4.68995 -2.80585     2243 <5e-04 ***
## as.factor(year)2012         -2.84605 -3.69267 -1.97081     2000 <5e-04 ***
## as.factor(year)2013         -4.63309 -5.77263 -3.50379     2000 <5e-04 ***
## as.factor(year)2014         -2.14923 -3.12542 -1.22036     2000 <5e-04 ***
## as.factor(year)2015         -1.26612 -2.40936 -0.07150     1978  0.036 *
## as.factor(year)2016         -0.44158 -1.26401  0.36510     2000  0.296
## as.factor(year)2017         -7.61699 -9.84913 -5.66562     1873 <5e-04 ***
## slope                       -0.20850 -1.82816  1.57488     2225  0.796
## as.factor(year)1988:slope    0.91210 -1.43018  3.26870     2000  0.416
## as.factor(year)1989:slope   -1.39392 -3.84775  0.65291     2211  0.211
## as.factor(year)1990:slope   -1.78905 -4.30133  0.45081     2260  0.142
## as.factor(year)1991:slope    0.35275 -1.97814  2.31326     2139  0.763
## as.factor(year)1992:slope    2.40453 -0.43973  5.45250     2000  0.110
## as.factor(year)1993:slope    0.13723 -2.15071  2.38213     2320  0.894
## as.factor(year)1994:slope   -1.72255 -4.48700  1.12516     2000  0.211
## as.factor(year)1995:slope    2.91365 -1.90461  7.37016     2000  0.220
## as.factor(year)1996:slope   -1.43853 -5.07471  2.73150     2000  0.467
## as.factor(year)2006:slope   -0.69183 -2.61421  1.40612     2183  0.504
## as.factor(year)2007:slope   -1.26947 -3.20753  0.54752     2249  0.174
```

```
## as.factor(year)2008:slope  -0.60969 -2.68121  1.13960     2210  0.531
## as.factor(year)2009:slope  -1.20472 -3.19644  0.95219     2000  0.262
## as.factor(year)2010:slope  -0.33109 -2.28498  1.51633     2175  0.764
## as.factor(year)2011:slope  -1.39078 -3.65783  0.75155     2296  0.189
## as.factor(year)2012:slope  -0.90318 -3.11035  0.96800     2246  0.372
## as.factor(year)2013:slope   0.82283 -1.48932  3.19521     2000  0.481
## as.factor(year)2014:slope  -0.65986 -2.69030  1.26655     2335  0.517
## as.factor(year)2015:slope   0.79913 -1.28538  3.16572     2143  0.487
## as.factor(year)2016:slope  -0.36577 -2.12764  1.60731     2210  0.705
## as.factor(year)2017:slope  -2.96409 -6.40327  0.08265     2000  0.060 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_years_abs_temp_bayes <- MCMCglmm(round(n_intact_seeds)~cmean_4*(intercept+slope),
                     random = ~ id,
                     rcov = ~units,
                     data = data_5yrs,
                     prior = prior_years,
                     family = "poisson",
                     nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_temp_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/obje
```

```r
summary(model_years_abs_temp_bayes)
```

Temp*(intercept+slope)

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3928.957
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id    0.5111   0.1773   0.8941     2360
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     5.949     5.06    6.849     2000
##
##  Location effects: round(n_intact_seeds) ~ cmean_4 * (intercept + slope)
##
##                     post.mean  l-95% CI  u-95% CI eff.samp pMCMC
## (Intercept)         -0.254782 -0.496052 -0.037612     2000 0.031 *
## cmean_4              0.001433 -0.129361  0.127174     2065 0.981
## intercept           -0.278305 -0.719653  0.154670     2000 0.209
## slope               -0.011347 -1.051083  1.148774     2317 0.982
```

```
## cmean_4:intercept  0.152642 -0.125917  0.412518     2000 0.269
## cmean_4:slope     -0.629363 -1.267200  0.046709     2000 0.062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(model_years_abs_temp_bayes) # High VIFs for intercept and slope
```

```
##           cmean_4         intercept             slope cmean_4:intercept
##          1.008389          8.131816          8.148382          7.866347
##     cmean_4:slope
##          7.881782
```

```r
model_years_abs_temp_slope_bayes <- MCMCglmm(round(n_intact_seeds)~cmean_4*slope,
                    random = ~ id,
                    rcov = ~units,
                    data = data_5yrs,
                    prior = prior_years,
                    family = "poisson",
                    nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_temp_slope_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code
```

```r
summary(model_years_abs_temp_slope_bayes)
```

Temp*slope

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3930.916
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id    0.4971    0.197   0.8868     2000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     5.919    5.081    6.885     2000
##
##  Location effects: round(n_intact_seeds) ~ cmean_4 * slope
##
##               post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## (Intercept)   -0.246557 -0.465170 -0.013734     2000  0.037 *
## cmean_4       -0.001892 -0.130093  0.125671     2312  0.962
## slope         -0.642919 -1.022182 -0.243454     2000 <5e-04 ***
## cmean_4:slope -0.280027 -0.532465 -0.024062     2000  0.024 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(model_years_abs_temp_slope_bayes)
```

```
##      cmean_4         slope cmean_4:slope
##     1.007184      1.019885      1.026908
```

```r
model_years_abs_temp_slope_vol_bayes <- MCMCglmm(round(n_intact_seeds)~cmean_4*slope+sqrt(shoot_vol), #
                     random = ~ id,
                     rcov = ~units,
                     data = subset(data_5yrs,shoot_vol>0), # There was one case with shoot_vol=0!
                     prior = prior_years,
                     family = "poisson",
                     nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_temp_slope_vol_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/
```

```r
summary(model_years_abs_temp_slope_vol_bayes)
```

Temp*slope+volume

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3669.297
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id    0.4419   0.1462   0.7741     2000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     5.581    4.756    6.456     2263
##
##  Location effects: round(n_intact_seeds) ~ cmean_4 * slope + sqrt(shoot_vol)
##
##                 post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)      -1.31877 -1.77713 -0.93373     1826 <5e-04 ***
## cmean_4           0.05276 -0.07782  0.17597     2000  0.415
## slope            -0.40115 -0.77517 -0.02339     1843  0.029 *
## sqrt(shoot_vol)   0.02667  0.01768  0.03541     2000 <5e-04 ***
## cmean_4:slope    -0.16200 -0.40551  0.08870     1820  0.207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(model_years_abs_temp_slope_vol_bayes)
```

```
##         cmean_4           slope sqrt(shoot_vol)   cmean_4:slope
##        1.017253        1.089447        1.093732        1.041041
```

```r
model_years_abs_slope_vol_bayes <- MCMCglmm(round(n_intact_seeds)~sqrt(shoot_vol)*slope, # Center shoot_
                        random = ~ id,
                        rcov = ~units,
                        data = subset(data_5yrs,shoot_vol>0),
                        prior = prior_years,
                        family = "poisson",
                        nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_slope_vol_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_ms2/code/
```

```r
summary(model_years_abs_slope_vol_bayes)
```

Volume*slope

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3670.687
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id    0.4212   0.1346   0.7311     1941
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     5.583    4.716    6.477     2000
##
##  Location effects: round(n_intact_seeds) ~ sqrt(shoot_vol) * slope
##
##                       post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## (Intercept)           -1.509888 -2.001098 -1.041404     2000 <5e-04 ***
## sqrt(shoot_vol)        0.032291  0.021250  0.042094     2000 <5e-04 ***
## slope                 -0.949709 -1.664310 -0.220060     2000  0.011 *
## sqrt(shoot_vol):slope  0.013435 -0.003564  0.027573     2000  0.096 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(model_years_abs_slope_vol_bayes)
```

```
##       sqrt(shoot_vol)                 slope sqrt(shoot_vol):slope
##              1.461460              4.528341              5.450202
```

```r
model_years_abs_temp_slope_vol_int_bayes <- MCMCglmm(round(n_intact_seeds)~cmean_4*slope+
                                          sqrt(shoot_vol)*slope, # Center shoot_vol?
                        random = ~ id,
                        rcov = ~units,
                        data = subset(data_5yrs,shoot_vol>0),
```

```r
                      prior = prior_years,
                      family = "poisson",
                      nitt = 2100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
# nitt = burnin + thin*(n samples to keep)
# Aim to store 2000 iterations
save(model_years_abs_temp_slope_vol_int_bayes,file="C:/Users/User/Dropbox/SU/Projects/lathyrus/lathyrus_
```

```r
summary(model_years_abs_temp_slope_vol_int_bayes)
```

Temp*slope+volume*slope

```
##
##  Iterations = 100001:2099001
##  Thinning interval  = 1000
##  Sample size  = 2000
##
##  DIC: 3667.916
##
##  G-structure:  ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id    0.4243   0.1299   0.7401     2171
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     5.606    4.776    6.579     2000
##
##  Location effects: round(n_intact_seeds) ~ cmean_4 * slope + sqrt(shoot_vol) * slope
##
##                       post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## (Intercept)           -1.490528 -1.956677 -1.010263     2000 <5e-04 ***
## cmean_4                0.060284 -0.064464  0.194921     1863  0.370
## slope                 -0.974881 -1.679461 -0.241139     2000  0.007 **
## sqrt(shoot_vol)        0.031867  0.021462  0.042378     2000 <5e-04 ***
## cmean_4:slope         -0.162925 -0.419596  0.088172     1829  0.208
## slope:sqrt(shoot_vol)  0.014734 -0.001006  0.030194     2000  0.058 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(model_years_abs_temp_slope_vol_int_bayes)
```

```
##               cmean_4                 slope       sqrt(shoot_vol)
##              1.022922              4.566252              1.472211
##         cmean_4:slope slope:sqrt(shoot_vol)
##              1.041201              5.482699
```

Model with relative fitness (not used)

```r
data_5yrs_means<-data_5yrs%>%group_by(year)%>%
  dplyr::summarise(n_intact_seeds_mean=mean(n_intact_seeds))
data_5yrs<-data_5yrs%>%left_join(data_5yrs_means,by="year")
```

```r
# Relativize fitness within years
data_5yrs$n_intact_seeds_rel_y<-with(data_5yrs,n_intact_seeds/n_intact_seeds_mean)
```

```r
model_years_rel<-glmmTMB(n_intact_seeds_rel_y~intercept+slope+
                           as.factor(year):intercept+as.factor(year):slope+(1|id),
                         subset(data_5yrs,year<2017),family=gaussian)
# Gaussian model, but fit should be bad
# summary(model_years_rel)
Anova(model_years_rel)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: n_intact_seeds_rel_y
##                             Chisq Df Pr(>Chisq)
## intercept                  2.6952  1    0.10065
## slope                      1.2430  1    0.26489
## intercept:as.factor(year) 24.0203 20    0.24151
## slope:as.factor(year)     28.6666 20    0.09453 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

TO DO

Things that I would like to do next, but I am unsure about how to modify the code (MCMCglmm models and priors) to achieve this:

- Include interaction between year and intercept/slope of the RN in the random part of the MCMCglmm models: to see if selection varies among years (now trying this by fitting yearly models) (To answer question 3)
- Include interaction between temperature and intercept/slope of the RN in the random part of the MCMCglmm models: to see if among-year variation in selection is related to temperature (To answer question 4)
- Include interaction between plant size (shoot volume) and intercept/slope of the RN 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)