Lathyrus ms2: Selection on reaction norms - multivariate modeling for phenotypic selection on plasticity 2a (Arnold et al. 2019 Phil. Trans. R. Soc. B)

Contents

Code based on Arnold et al. 2019 Phil. Trans. R. Soc. B. Using plant individuals with at least 4 years of data.

Stack data

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

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

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

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

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

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

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

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

```
##   Obs id year temp fitness.FFD.stack  traits variable   family
## 1   1  1 2006    0         4.1949956 fitness  fitness gaussian
## 2   2  2 2007    0         1.2394190 fitness  fitness gaussian
## 3   3  3 2007    0         0.7171445 fitness  fitness gaussian
## 4   4  4 2007    0         1.7884822 fitness  fitness gaussian
## 5   5  5 2007    0         1.1468928 fitness  fitness gaussian
## 6   6  6 2007    0         0.5049888 fitness  fitness gaussian
```

Univariate models

Fitness with no random effects.

Priors are used to keep the MCMCglmm model fitting process within appropriate parameter space. For the purpose of this code, we only need to specify the distributions of residuals (R) and random effects (G) based on the number of model variables and their (co)variances (V).

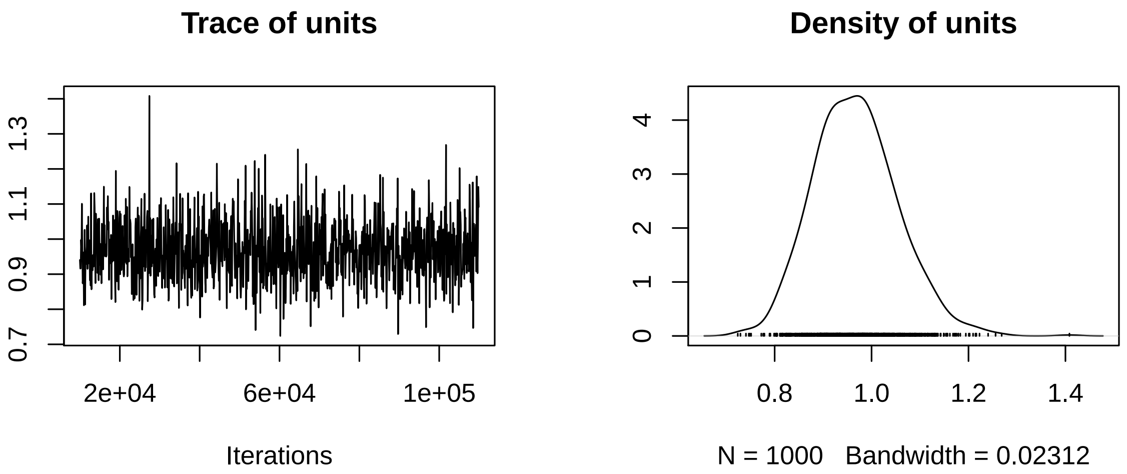
```
# Scaling factor for MCMCglmm iterations
sc <- 100#0 # Increase this parameter for longer runs

priorUV0 <- list(R = list(R1 = list(V = diag(1), nu = 2)))

univar.fitness <- MCMCglmm(mean_fitness_rel ~ 1,
                          data = data_4yrs_total,
                          prior = priorUV0,
                          family = "gaussian",
                          nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
```

```
summary(univar.fitness)
```

```
##
##  Iterations = 10001:109901
##  Thinning interval  = 100
##  Sample size  = 1000
##
##  DIC: 682.1289
##
##  R-structure:  ~units
##
```

```
##       post.mean l-95% CI u-95% CI eff.samp
## units    0.9674   0.8107    1.135     1000
##
##  Location effects: mean_fitness_rel ~ 1
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)    1.0035   0.8813   1.1212     1000 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(univar.fitness$VCV)
```



FFD with random effects of year and individual-intercept

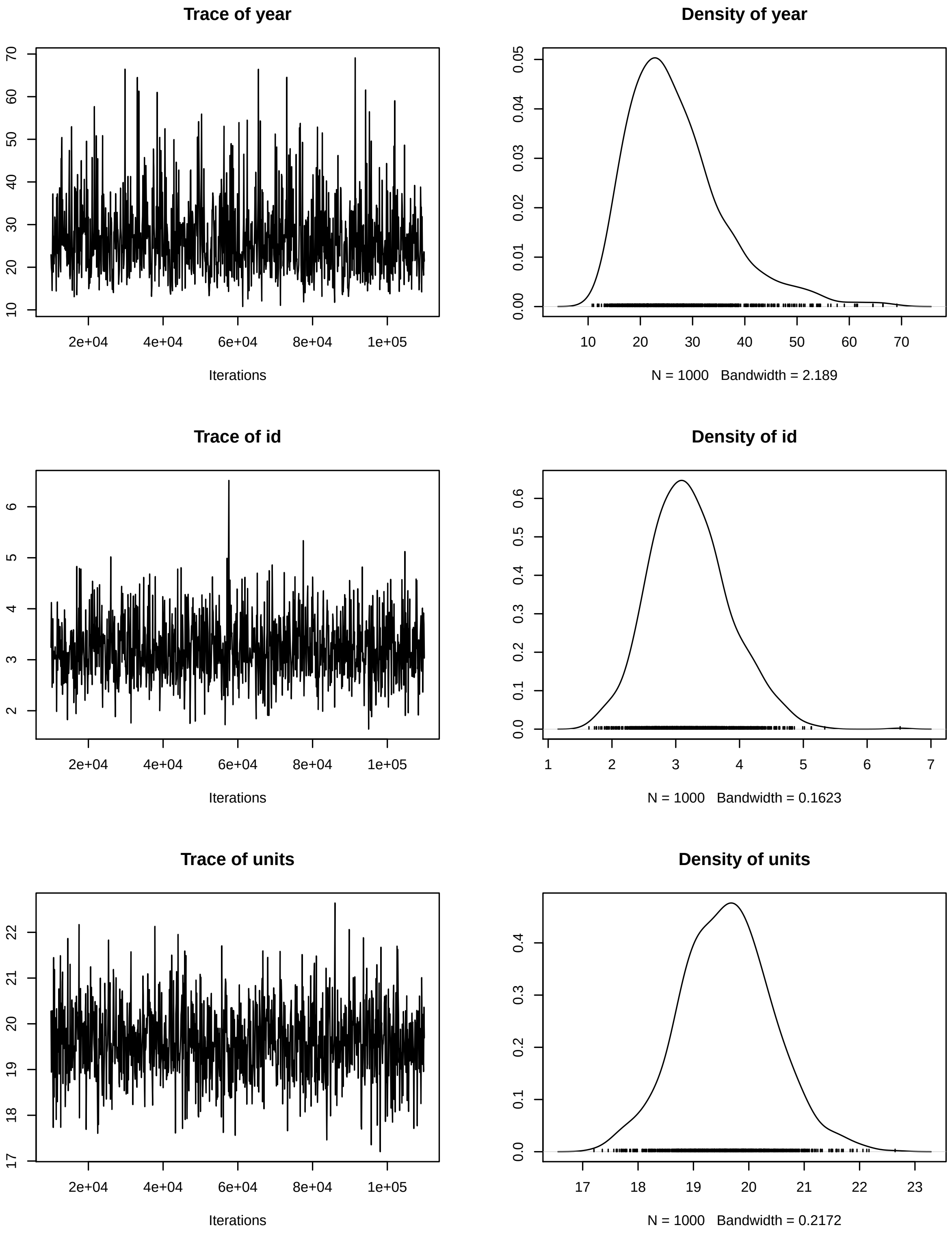
```
priorUV2 <- list(G = list(G1 = list(V = diag(1), nu = 1),  # for random effect of year
                          G2 = list(V = diag(1), nu = 1)), # for random effect of id
                 R = list(R1 = list(V = diag(1), nu = 2)))

univar.FFD <- MCMCglmm(FFD ~ cmean_4,
                     random = ~year + id,
                     rcov = ~units,
                     data = data_4yrs,
                     prior = priorUV2,
                     family = "gaussian",
                     nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
```

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

```
##
##  Iterations = 10001:109901
##  Thinning interval  = 100
##  Sample size  = 1000
##
##  DIC: 8599.696
##
##  G-structure:  ~year
```

```
##
##      post.mean l-95% CI u-95% CI eff.samp
## year     26.93    13.16    46.51     1000
##
##                 ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id     3.221    2.159     4.63     1000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units      19.6    17.91    21.19     1000
##
##  Location effects: FFD ~ cmean_4
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   58.0492  55.5045  59.9546    909.3 <0.001 ***
## cmean_4       -2.4444  -3.9852  -0.8239   1145.8  0.006 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(univar.FFD$VCV)
```


Trace of year
Iterations
Density of year
N = 1000 Bandwidth = 2.189
Trace of id
Iterations
Density of id
N = 1000 Bandwidth = 0.1623
Trace of units
Iterations
Density of units
N = 1000 Bandwidth = 0.2172
2e+04
4e+04
6e+04
8e+04
1e+05

Random regression for FFD, including random effects of individual slopes

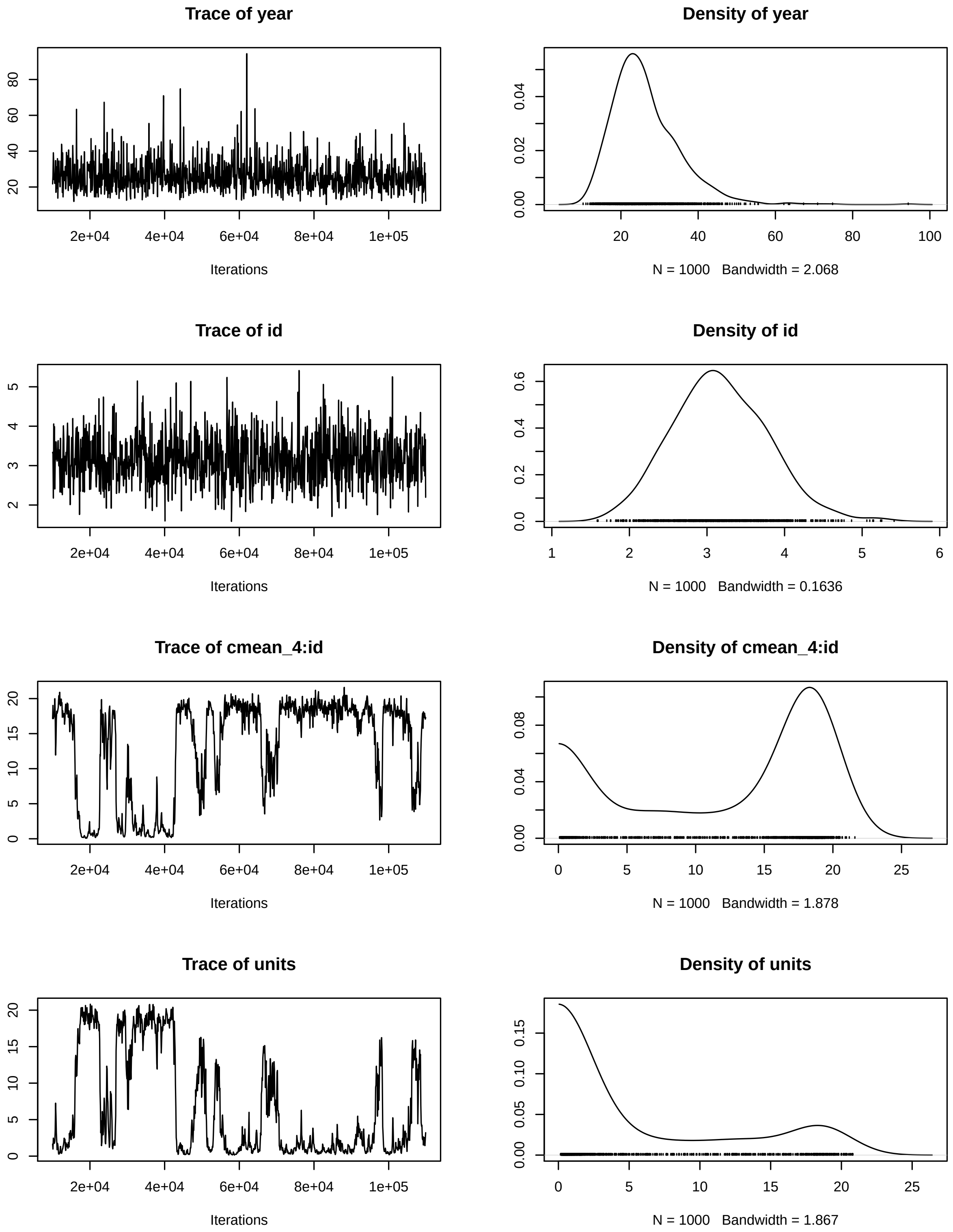
With variance in intercepts and slopes but no covariance (only two random effects).

```
priorUV3_NoCov <- list(G = list(G1 = list(V = diag(1), nu = 1),
                                # other random effect (YEAR)
                                G2 = list(V = diag(1), nu = 1),  # variance in intercepts
                                G3 = list(V = diag(1), nu = 1)), # variance in slopes
                       R = list(R1 = list(V = diag(1), nu = 2)))

# This model runs, but obviously does not have a covariance between intercept and slope
univar.FFD_RR_noCov <- MCMCglmm(FFD ~ cmean_4,
                                random = ~year + id + cmean_4:id,
                                rcov = ~units,
                                data = data_4yrs,
                                prior = priorUV3_NoCov,
                                family = "gaussian",
                                nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
```

```
summary(univar.FFD_RR_noCov)
```

```
##
##  Iterations = 10001:109901
##  Thinning interval  = 100
##  Sample size  = 1000
##
##  DIC: 5563.329
##
##  G-structure:  ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year     26.39    12.16    42.65    945.5
##
##                ~id
##
##    post.mean l-95% CI u-95% CI eff.samp
## id     3.167    1.917    4.275     1098
##
##                ~cmean_4:id
##
##            post.mean l-95% CI u-95% CI eff.samp
## cmean_4:id     13.06   0.1775    19.87    12.52
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     6.576   0.1824     19.3    11.25
##
##  Location effects: FFD ~ cmean_4
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   58.0905  55.9173  60.0772     1000 <0.001 ***
## cmean_4       -2.4151  -3.9895  -0.9043     1000  0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(univar.FFD_RR_noCov$VCV)
```



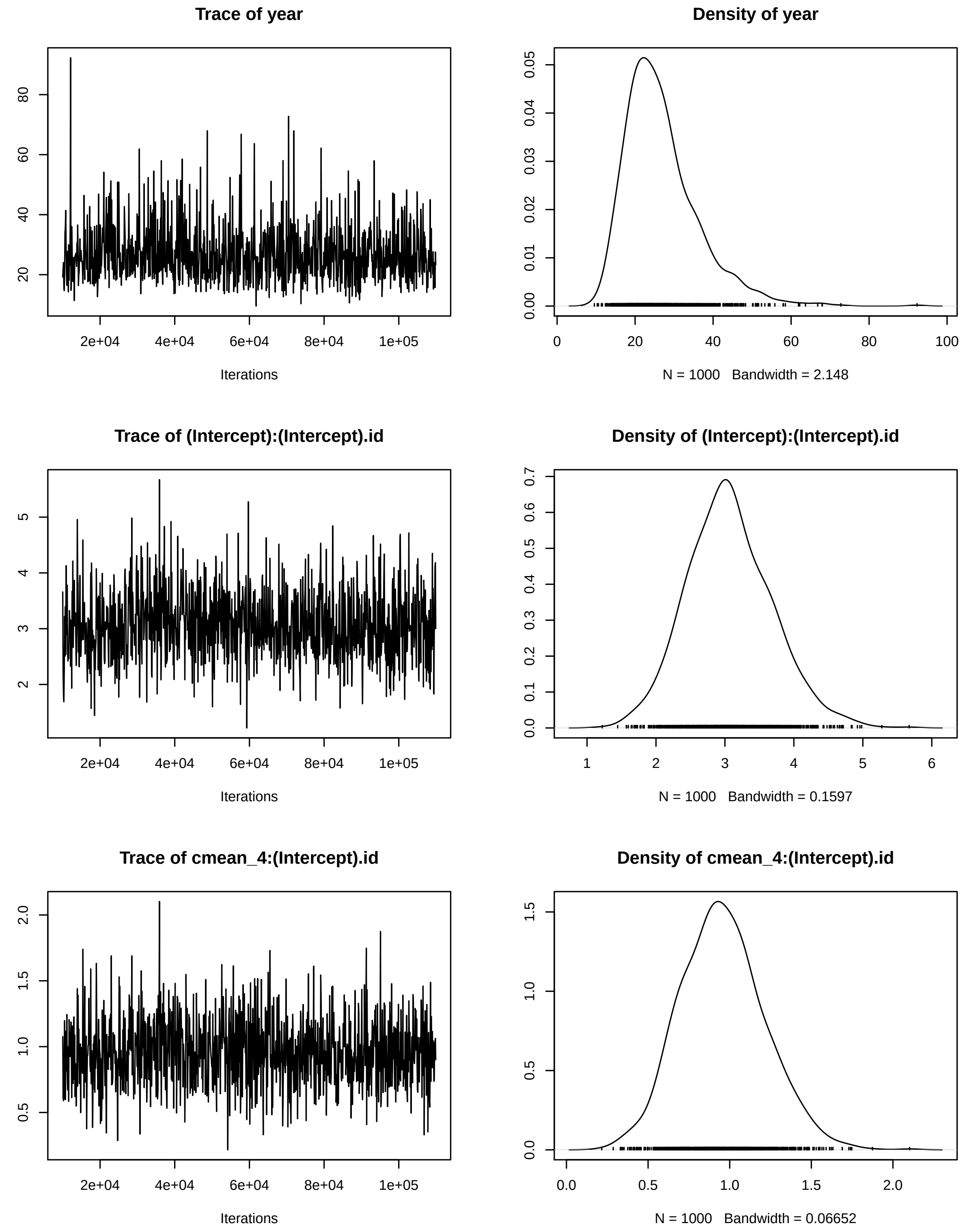
Including covariance between intercept and slope

```r
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)))

univar.FFD_RR <- MCMCglmm(FFD ~ cmean_4,
                        random = ~year + us(1 + cmean_4):id,
                        rcov = ~units,
                        data = data_4yrs,
                        prior = priorUV2_RR,
                        family = "gaussian",
                        nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
```

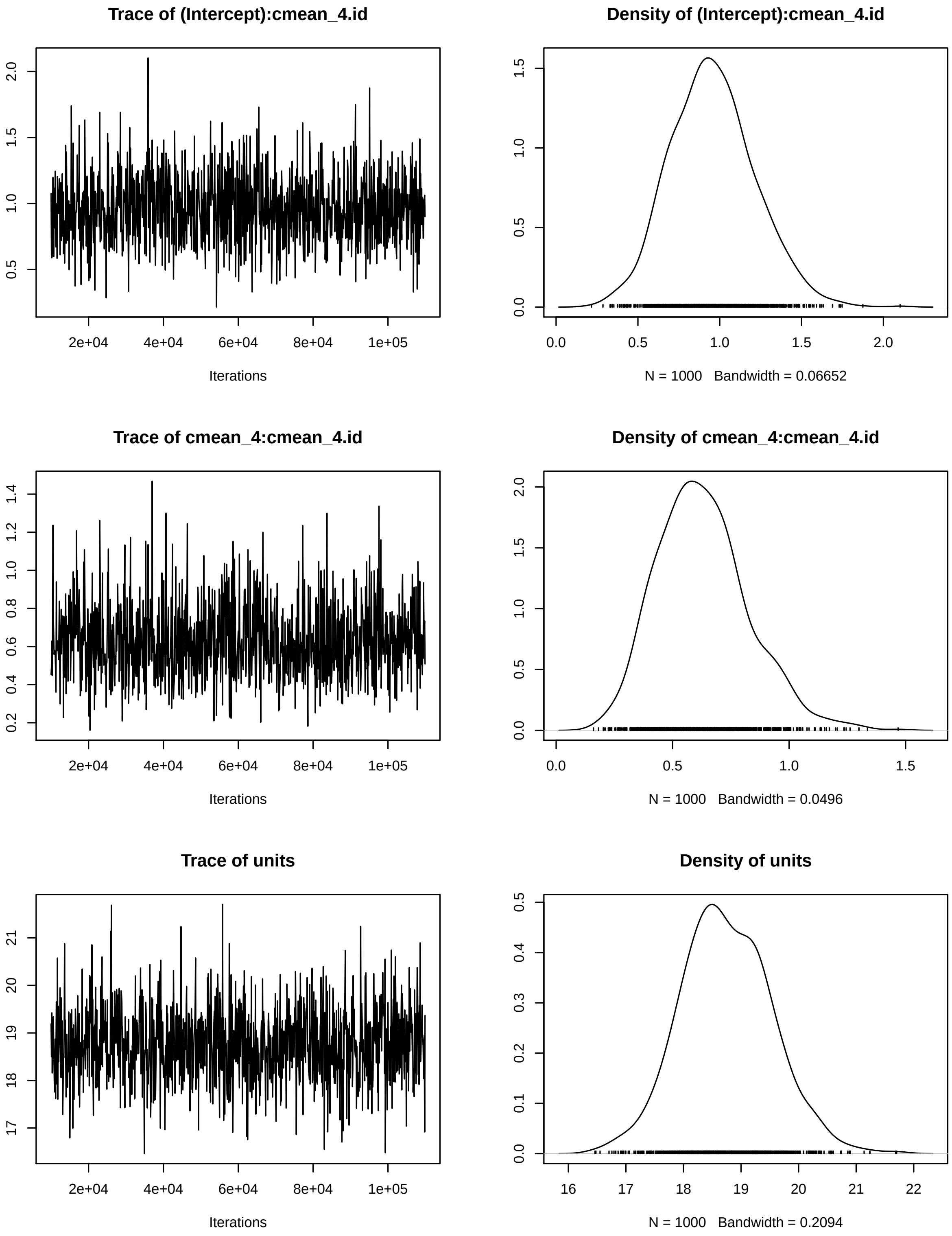
```r
summary(univar.FFD_RR)
```

```
##
##  Iterations = 10001:109901
##  Thinning interval  = 100
##  Sample size  = 1000
##
##  DIC: 8564.618
##
##  G-structure:  ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year     26.88    13.54    47.13     1000
##
##                 ~us(1 + cmean_4):id
##
##                             post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).id    3.0670   1.9565    4.351     1000
## cmean_4:(Intercept).id        0.9570   0.4952    1.488     1000
## (Intercept):cmean_4.id        0.9570   0.4952    1.488     1000
## cmean_4:cmean_4.id            0.6291   0.2696    1.003     1000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     18.74    17.25    20.31    920.7
##
##  Location effects: FFD ~ cmean_4
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   58.1070  55.7936  59.9426    887.5 <0.001 ***
## cmean_4       -2.4058  -4.1688  -0.9526   1000.0 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(univar.FFD_RR$VCV[,1:3])
```



```
plot(univar.FFD_RR$VCV[,4:6])
```



(Intercept):(Intercept).id represents among-individual variation in intercepts.

cmean_4:(Intercept).id and (Intercept):cmean_4.id represent the covariance between intercept and slope.

cmean_4:cmean_4.id represents the among-individual variation in slopes.

There is among-individual variation in plasticity (IxE) — i.e., individuals change their phenology at different rates in response to temperature.

For interpretation of the among-individual covariance between intercepts and slopes for FFD, it is often easier to convert this to a correlation. We can use the formula for a correlation with the posterior distributions of our (co)variance components, giving us a distribution of correlation values that we can use to calculate estimates and 95% credible intervals:

```
cor_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"]))
posterior.mode(cor_univar.FFD_RR_intslope)
```

```
##     var1
## 0.779509
```

```
HPDinterval(cor_univar.FFD_RR_intslope)
```

```
##          lower     upper
## var1 0.4483434 0.9130105
## attr(,"Probability")
## [1] 0.95
```

We find a strong positive correlation between among-individual variance in intercepts and slopes, at the intercept (x = 0).

Until here, this is similar to code in Houslay & Wilson 2017.

Bivariate models of both fitness and FFD

Having established that there is among-individual variation in plasticity, we want to now test whether there is an association between variation in FFD (intercept and/or slope) and fitness. We build bivariate models with two response variables, but only one of them (FFD) has repeated observations at the individual level and is a function of the x variable (temperature).

Random intercepts for individuals

First build a bivariate model with just random intercepts for individuals (id) for FFD. Specify a bivariate prior for MCMCglmm, with the additional random effect of year (as before) and then a random effect for individual. This is specified in the R section of the prior, because we want it to covary with the residual for fitness.

We estimate random intercepts of both traits (FFD and fitness).

The at.level keyword specifies effects as relating to only one of our response variables.

Houslay & Wilson 2017 standardised FFD by scaling it in order to make the multivariate model easier to fit. Here, it is not standardised.

```
priorBiv <- list(G = list(G1 = list(V = diag(1), nu = 1)),
                 # ^ random effect for year (fitted for FFD only)
                 R = list(R1 = list(V = diag(2), nu = 2, covu = TRUE),
                          # ^ 2-way var-cov matrix of id for FFD, residual for fitness
                          R2 = list(V = diag(1), nu = 1))) # residual for FFD

modelBV <- MCMCglmm(fitness.FFD.stack ~ variable - 1 +
                      # ^ fits a mean 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")):id,
                    # ^ random intercepts for individual
                    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.stack,
                    prior = priorBiv,
                    family = NULL, # specified already in the data-set
                    nitt = 1100 * sc, thin = sc, burnin = 100 * sc, verbose = F)
```

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

Table 1: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	58.149	56.207	60.381	1000	0.001
variablefitness	1.001	0.879	1.113	1000	0.001
at.level(variable, “FFD”):temp	-2.392	-3.840	-0.517	1000	0.012

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

Table 2: Random effects

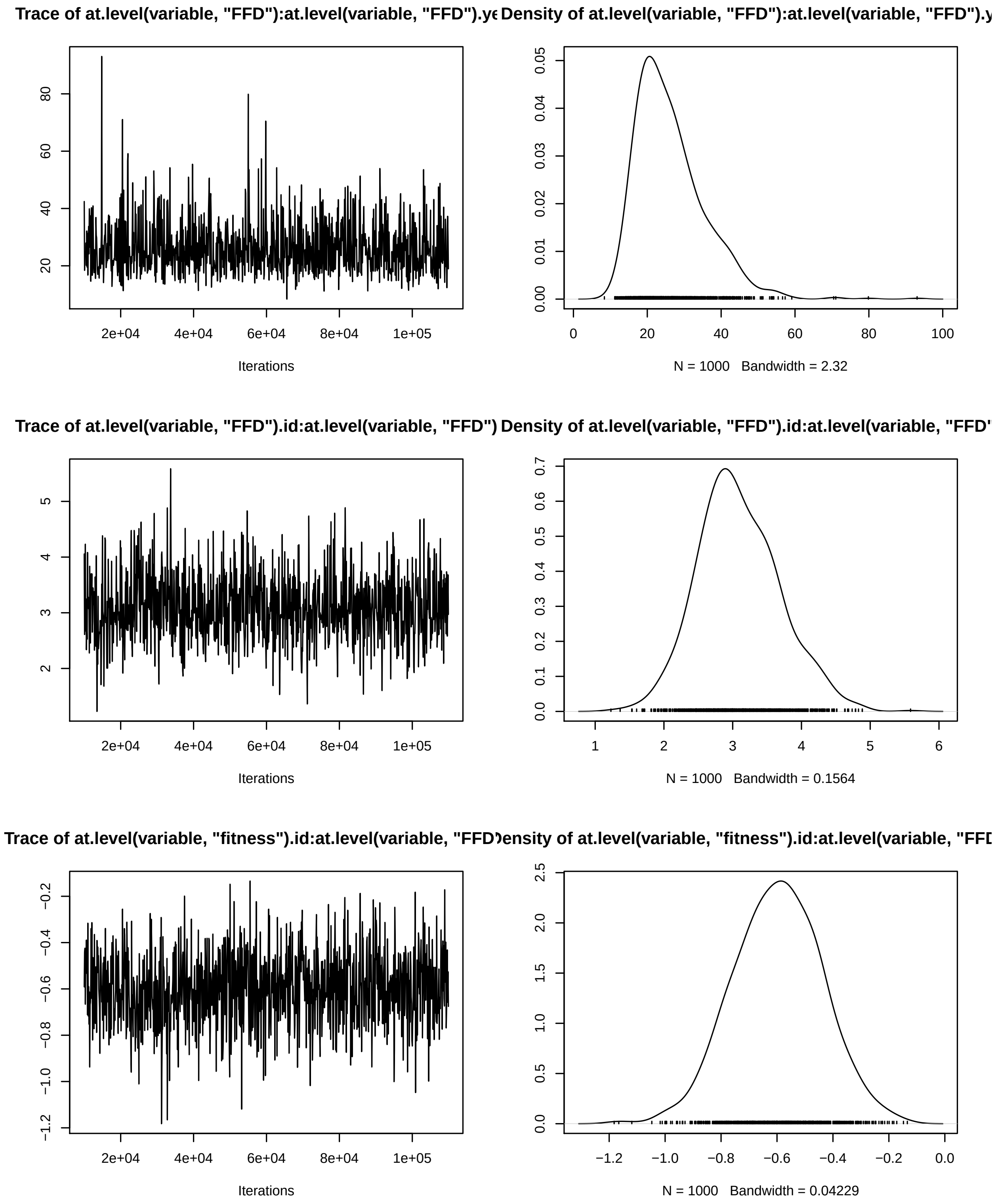
	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”):at.level(variable, “FFD”).year	26.301	11.174	43.256	1000

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

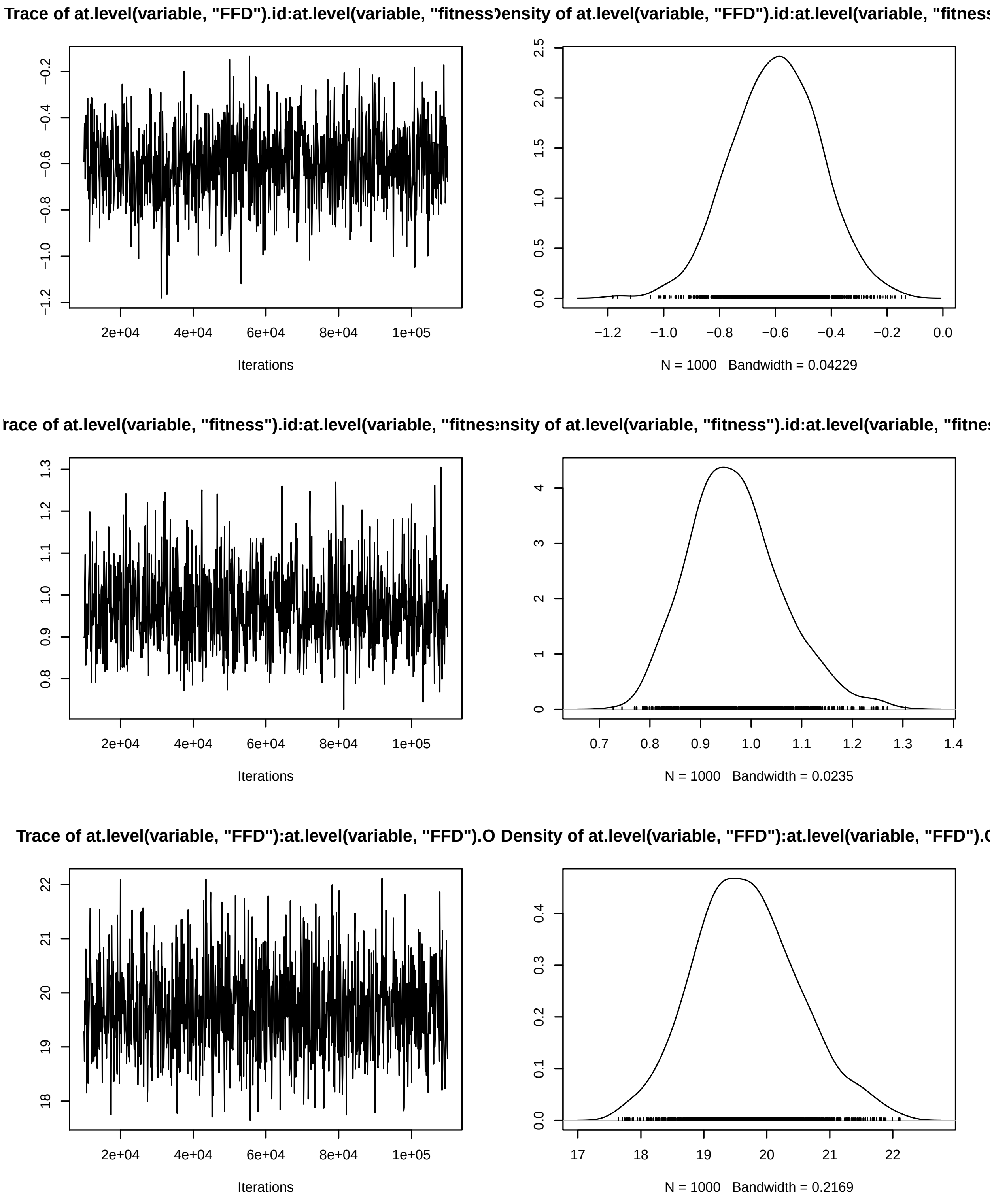
Table 3: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	3.090	2.003	4.325	888
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.598	-0.908	-0.299	1000
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.598	-0.908	-0.299	1000
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	0.969	0.788	1.138	1000
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	19.684	18.148	21.438	1419

```
plot(modelBV$VCV[,1:3])
```



```r
plot(modelBV$VCV[,4:6])
```


Trace of at.level(variable, "FFD").id:at.level(variable, "fitness
Density of at.level(variable, "FFD").id:at.level(variable, "fitness
−1.2
−1.0
−0.8
−0.6
−0.4
−0.2
2e+04
4e+04
6e+04
8e+04
1e+05
Iterations
0.0
0.5
1.0
1.5
2.0
2.5
−1.2
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
N = 1000 Bandwidth = 0.04229
race of at.level(variable, "fitness").id:at.level(variable, "fitnes
ensity of at.level(variable, "fitness").id:at.level(variable, "fitne
0.8
0.9
1.0
1.1
1.2
1.3
2e+04
4e+04
6e+04
8e+04
1e+05
Iterations
0
1
2
3
4
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
N = 1000 Bandwidth = 0.0235
Trace of at.level(variable, "FFD"):at.level(variable, "FFD").O
Density of at.level(variable, "FFD"):at.level(variable, "FFD").
18
19
20
21
22
2e+04
4e+04
6e+04
8e+04
1e+05
Iterations
0.0
0.1
0.2
0.3
0.4
17
18
19
20
21
22
N = 1000 Bandwidth = 0.2169

Random regressions for individuals (FINAL MODELS)

The final stage: Now build a bivariate model with random regressions for individuals across temperatures. Set MCMCglmm bivariate prior, with one additional random effect (for year) and random effects for individual (id) and individual slope (temp:id), which covary with the residual for fitness.

We estimate random intercepts of both traits (FFD and fitness), and random slopes for trait 1 only (FFD).

Houslay & Wilson 2017 standardised FFD by scaling it in order to make the multivariate model easier to fit. Here, it is not standardised. They also had random intercepts for both response traits: ~ us(at.level(trait,1)):year + us(trait + cmean_4:at.level(trait,1)):id Having random intercepts for fitness means that there is variance in intercepts for relative fitness - don't think we need that - we consider the ariance between indivdiuals in fitness as residual variance. They also used rcov =~ idh(trait):units. Both idh and us fit different variances across each component in formula, but us will also fit the covariances.

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

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

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

Table 4: Fixed effects

	post.mean	l-95% CI	u-95% CI	eff.samp	pMCMC
variableFFD	58.096	55.848	60.266	1000	0.001
variablefitness	1.000	0.860	1.112	1000	0.001
at.level(variable, “FFD”):temp	-2.402	-3.908	-0.770	1772	0.004

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

Table 5: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”):at.level(variable, “FFD”).year	26.416	11.955	46.132	1000

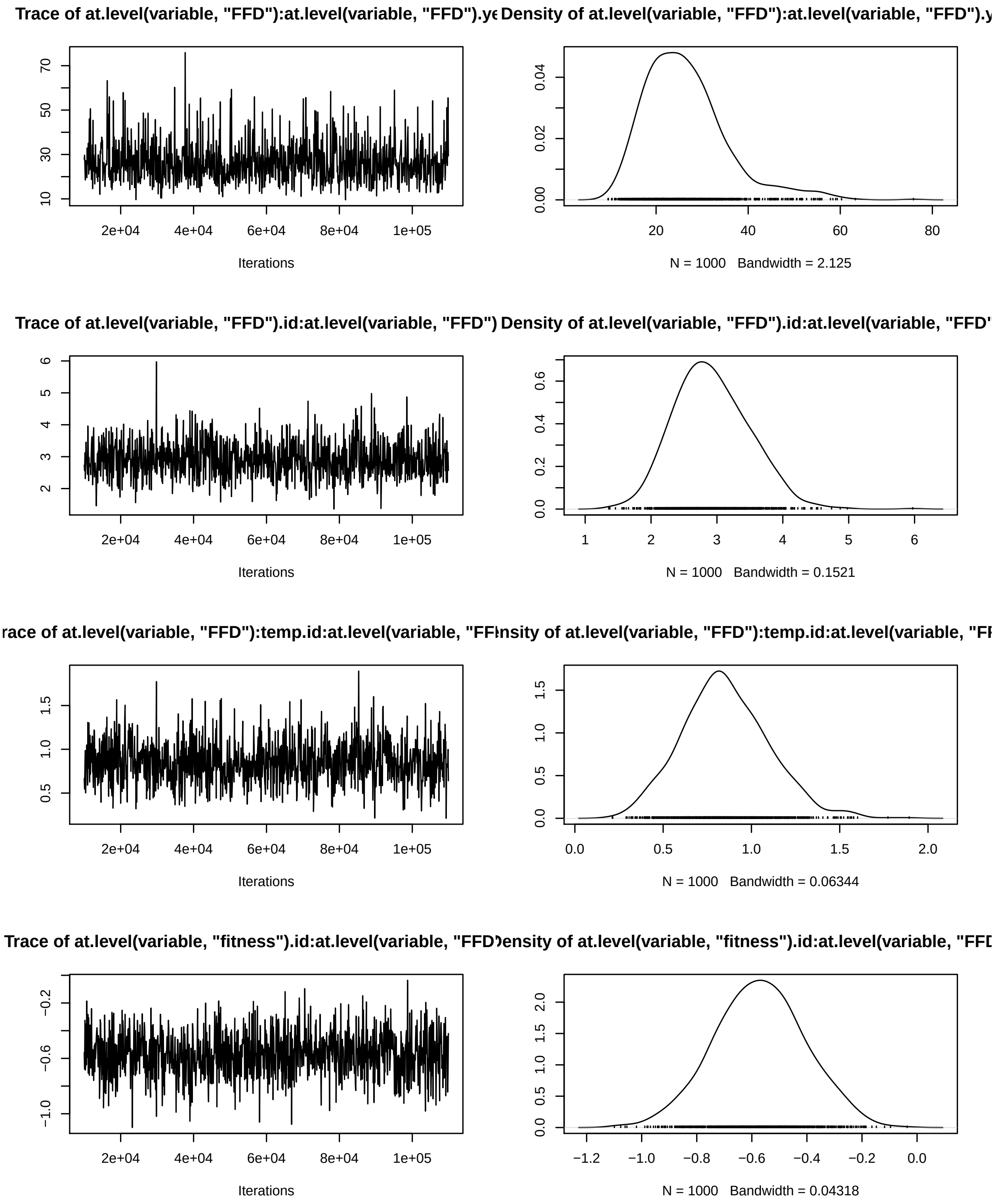
```
kable(summary(modelBV_RR)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 6: Random effects

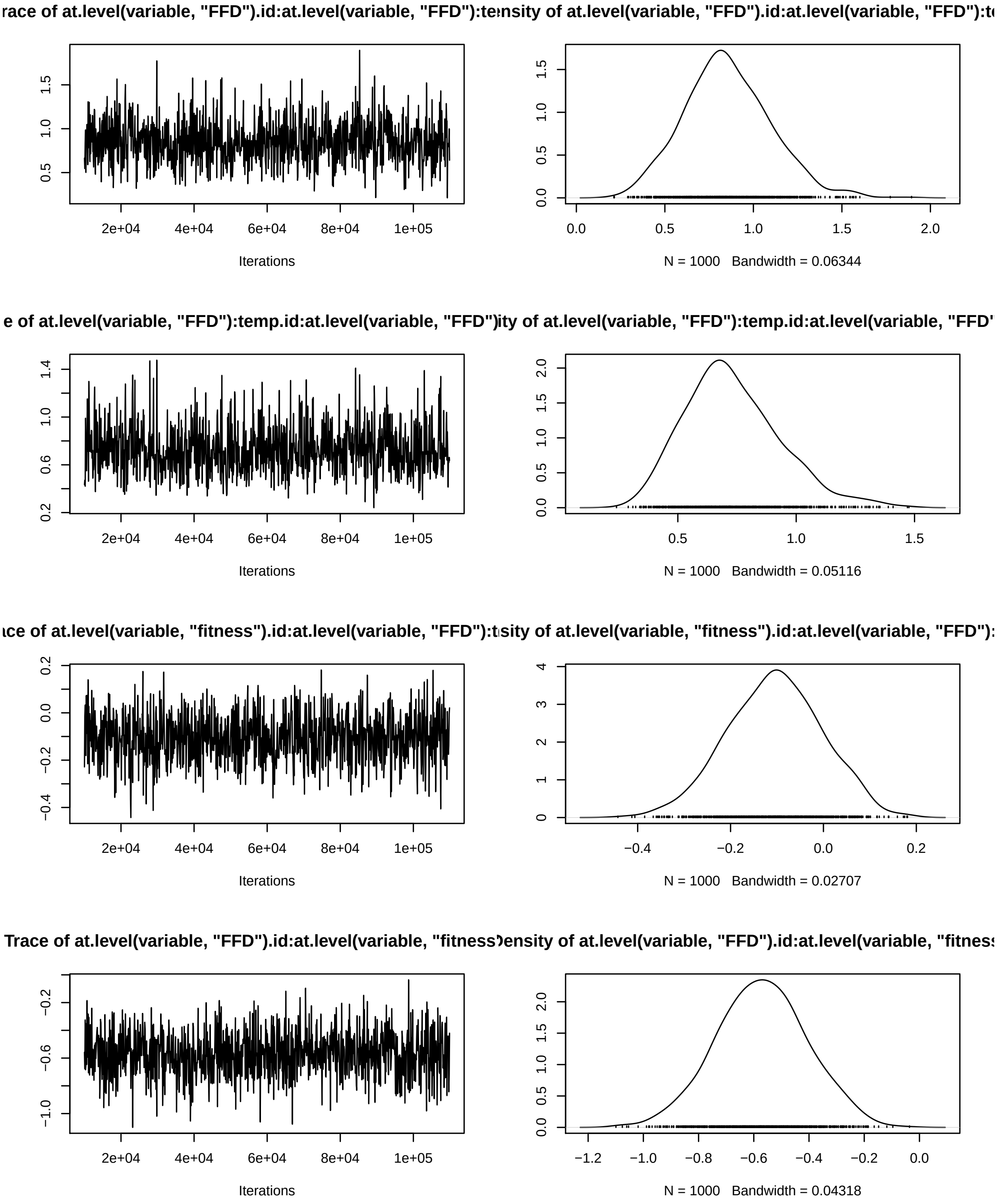
	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	2.924	1.971	4.048	1000
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	0.853	0.395	1.321	646
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.572	-0.872	-0.236	1148
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	0.853	0.395	1.321	646
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	0.731	0.343	1.103	907
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	-0.105	-0.303	0.082	1000
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.572	-0.872	-0.236	1148
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	-0.105	-0.303	0.082	1000
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	0.978	0.805	1.158	1000
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	18.714	17.178	20.213	1000

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

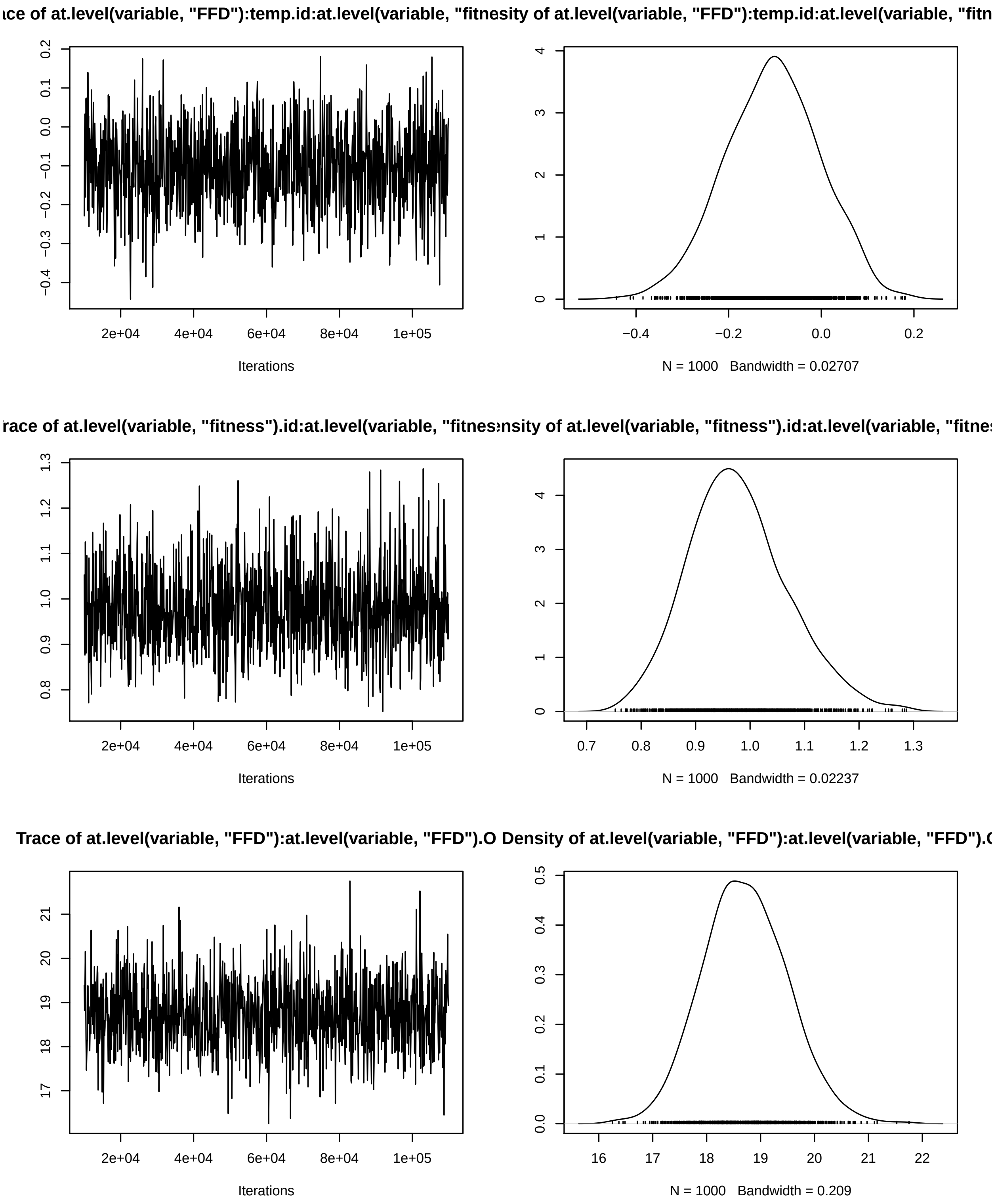
```
plot(modelBV_RR$VCV[,1:4])
```



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



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



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

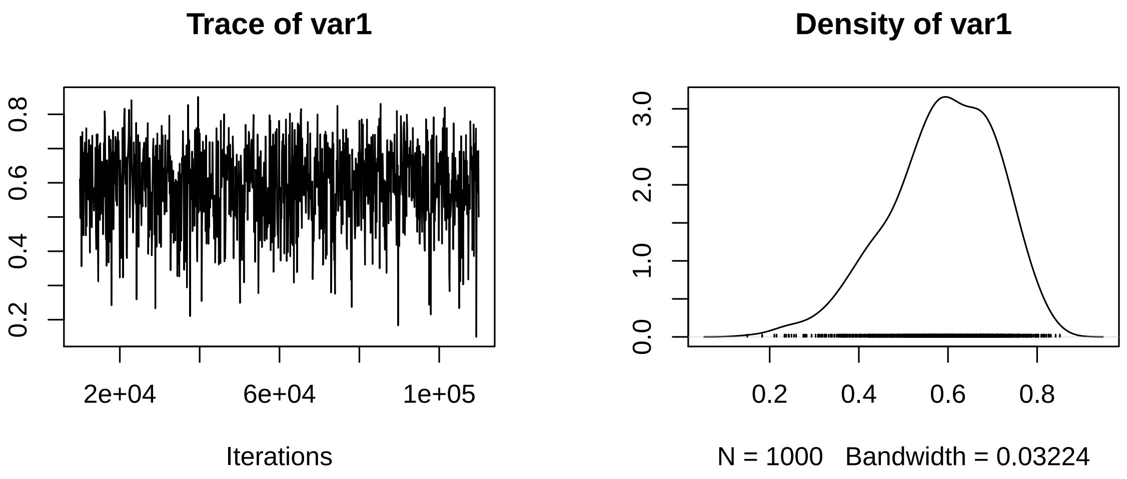
```
kable(diag(autocorr(modelBV_RR$VCV)[2, , ]),caption="Autocorrelation")
```

Table 7: Autocorrelation

	x
at.level(variable, “FFD”):at.level(variable, “FFD”).year	-0.0071223
at.level(variable, “FFD”).id:at.level(variable, “FFD”).id	0.0082818
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”).id	-0.0187058
at.level(variable, “fitness”).id:at.level(variable, “FFD”).id	-0.0695727
at.level(variable, “FFD”).id:at.level(variable, “FFD”):temp.id	-0.0187058
at.level(variable, “FFD”):temp.id:at.level(variable, “FFD”):temp.id	0.0481988
at.level(variable, “fitness”).id:at.level(variable, “FFD”):temp.id	-0.0005915
at.level(variable, “FFD”).id:at.level(variable, “fitness”).id	-0.0695727
at.level(variable, “FFD”):temp.id:at.level(variable, “fitness”).id	-0.0005915
at.level(variable, “fitness”).id:at.level(variable, “fitness”).id	-0.0145873
at.level(variable, “FFD”):at.level(variable, “FFD”).Obs	0.0267009

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

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



```
posterior.mode(cor_BV_RR_intslope)
```

```
##      var1
## 0.6738702
```

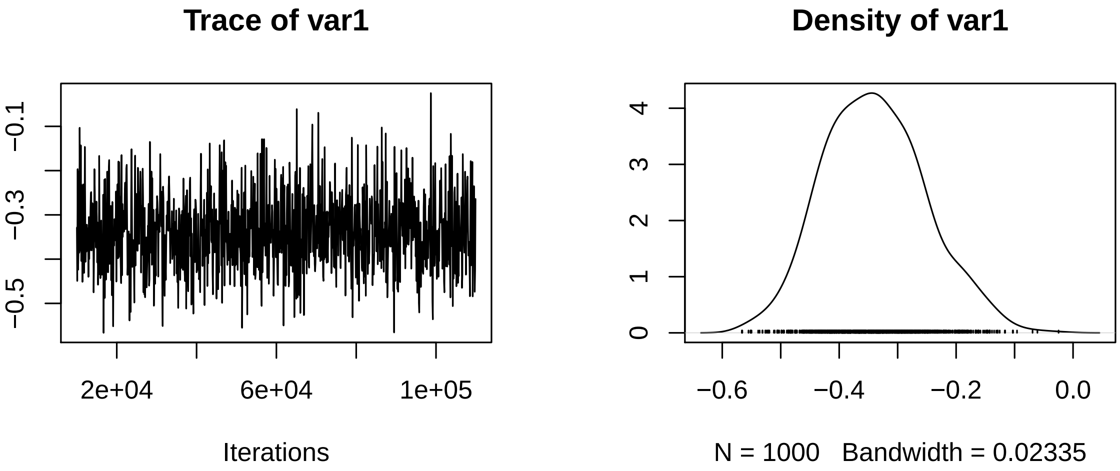
```
HPDinterval(cor_BV_RR_intslope)
```

```
##          lower     upper
## var1 0.3573666 0.8029384
## attr(,"Probability")
## [1] 0.95
```

We find a strong positive correlation between among-individual variance in intercepts and slopes, at the intercept (x = 0). Although it is a bit lower than in our earlier univariate random regression model (OK?).

Determining the among-individual correlation between FFD and fitness:

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



```
posterior.mode(cor_BV_RR_intfit)
```

```
##       var1
## -0.3330387
```

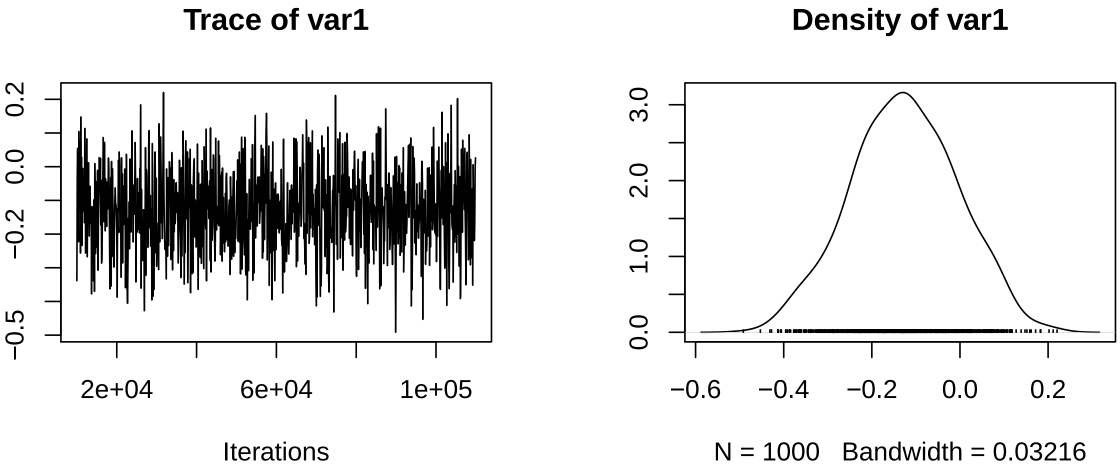
```
HPDinterval(cor_BV_RR_intfit)
```

```
##           lower      upper
## var1 -0.4988313 -0.1584265
## attr(,"Probability")
## [1] 0.95
```

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

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

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



```
posterior.mode(cor_BV_RR_slopefit)
```

```
##       var1
## -0.1292325
```

```
HPDinterval(cor_BV_RR_slopefit)
```

```
##           lower    upper
## var1 -0.3647857 0.097943
## attr(,"Probability")
## [1] 0.95
```

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

Extract selection coefficients from bivariate model with random regressions for individuals

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

```
##            [,1]        [,2]        [,3]
## [1,]  2.7054083  0.80951780 -0.57514498
## [2,]  0.8095178  0.71911350 -0.09519919
## [3,] -0.5751450 -0.09519919  0.93469360
```

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

```
## [1] -0.57514498 -0.09519919
```

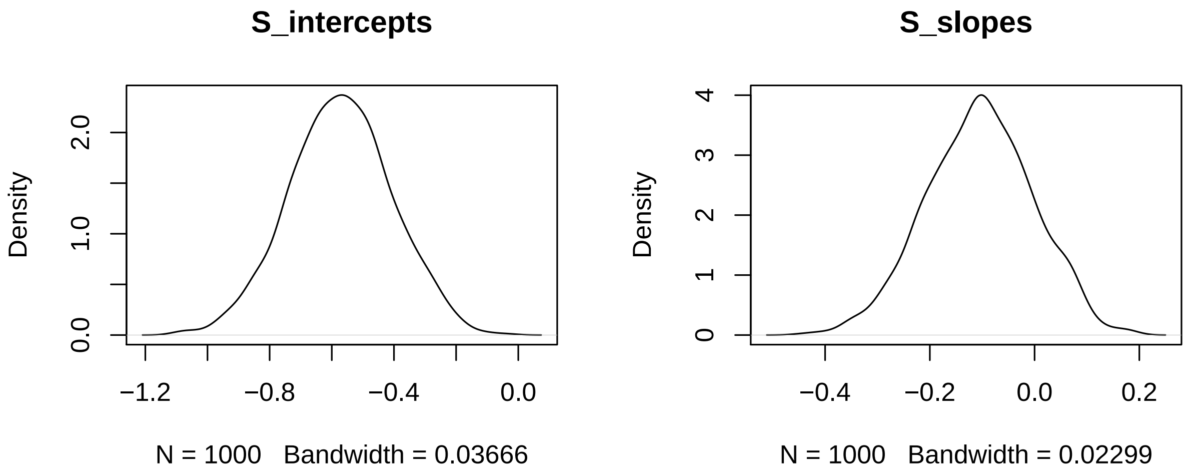
```r
posterior.mode(mcmc(S.modelBV_RR))
```

```
## S_intercepts     S_slopes
##  -0.57514498  -0.09519919
```

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

```
##                   lower       upper
## S_intercepts -0.8717273 -0.23643278
## S_slopes     -0.3026539  0.08238731
## attr(,"Probability")
## [1] 0.95
```

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



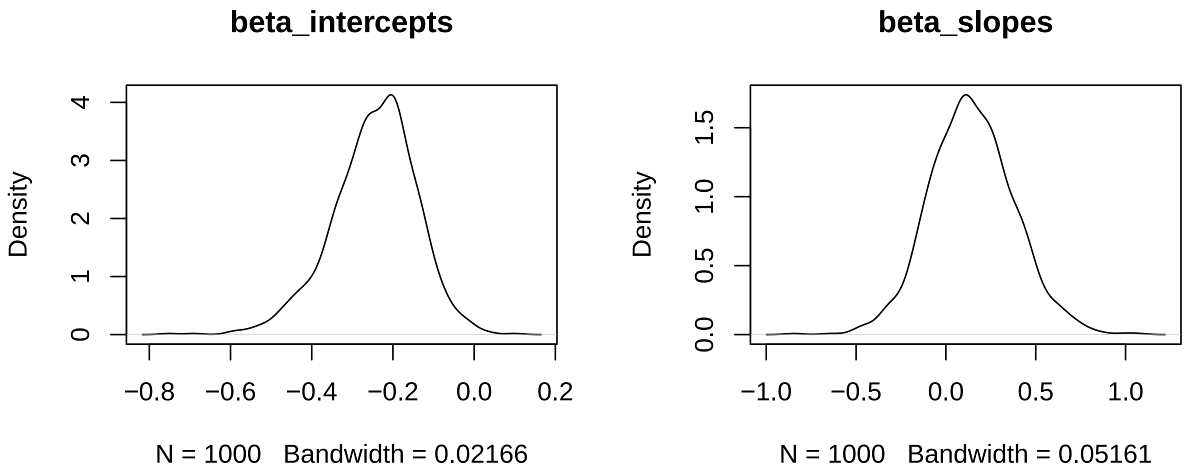
```r
# Estimate selection gradients for intercept and slope (beta = S / P)
# on each sample of posterior and extract their mode
n <- length(modelBV_RR$VCV[,2])   # sample size
beta_post_RR <- matrix(NA, n ,2)

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

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

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

```
## beta_intercepts     beta_slopes
##      -0.1984145       0.1044945
```

```
HPDinterval(mcmc(beta_post_RR))
```

```
##                      lower       upper
## beta_intercepts -0.4787639 -0.06983343
## beta_slopes     -0.3534181  0.57483387
## attr(,"Probability")
## [1] 0.95
```

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



beta_intercepts

beta_slopes