

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

Contents

Stack data	1
Univariate models	3
Fitness with no random effects.	3
FFD with random effects of year and individual-intercept	4
Random regression for FFD, including random effects of individual slopes	7
With variance in intercepts and slopes but no covariance (only two random effects).	7
Including covariance between intercept and slope	9
Bivariate models of both fitness and FFD	12
Random intercepts for individuals	12
Random regressions for individuals (FINAL MODELS)	16
Extract selection coefficients from bivariate model with random regressions for individuals	21

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

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(3,8)]%>%
    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

# If you like, view the stacked data file
View(data.stack)

```

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

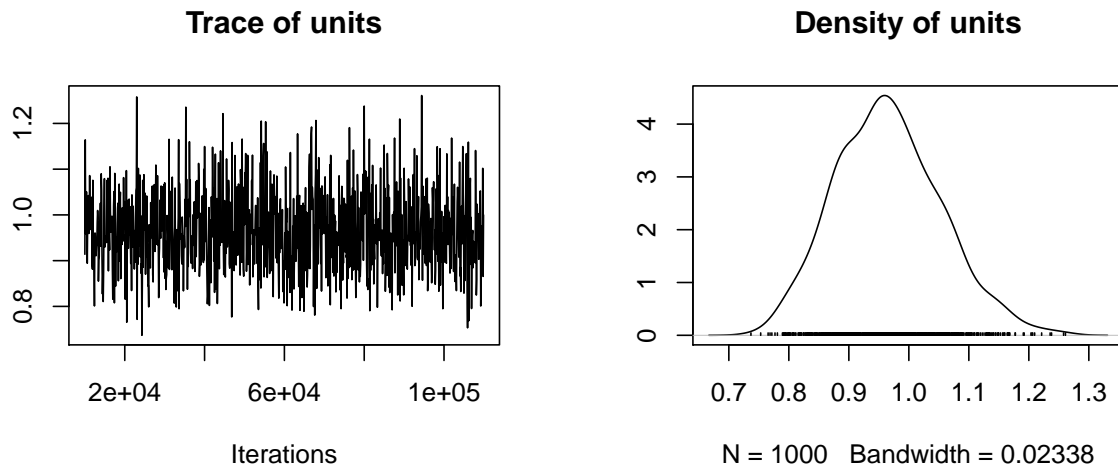
priorUV0 <- list(R = list(R1 = list(V = diag(1), nu = 2)))

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: 682.1497
##
## R-structure: ~units
##
##      post.mean l-95% CI u-95% CI eff.samp
## units    0.9654  0.7948    1.134    1491
##
## Location effects: mean_fitness_rel ~ 1
##
##      post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)  0.9979  0.8707    1.1236    1126 <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)

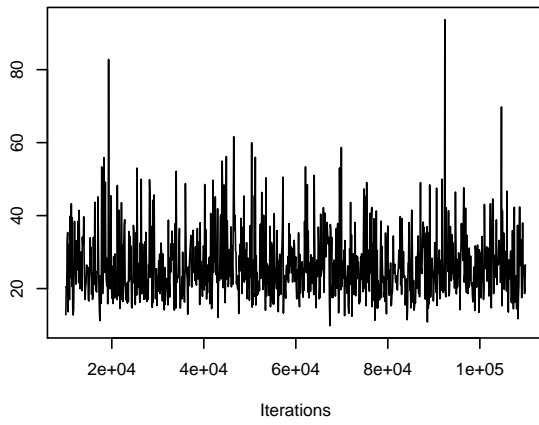
summary(univar.FFD)
```

```
##
## Iterations = 10001:109901
## Thinning interval = 100
## Sample size = 1000
##
## DIC: 8599.951
##
## G-structure: ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year      26.43   12.75   45.47     1000
##
##      ~id
##
##      post.mean l-95% CI u-95% CI eff.samp
## id       3.166    1.975    4.384     1103
##
## R-structure: ~units
##
```

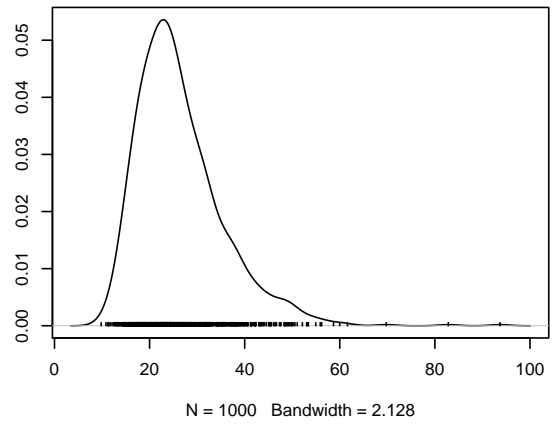
```
##           post.mean l-95% CI u-95% CI eff.samp
## units      19.66    17.98    21.18    1000
##
## Location effects: FFD ~ cmean_4
##
##           post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)   58.133   55.771   60.007    1000 <0.001 ***
## cmean_4       -3.233   -5.397   -1.016    1000  0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot(univar.FFD$VCV)
```

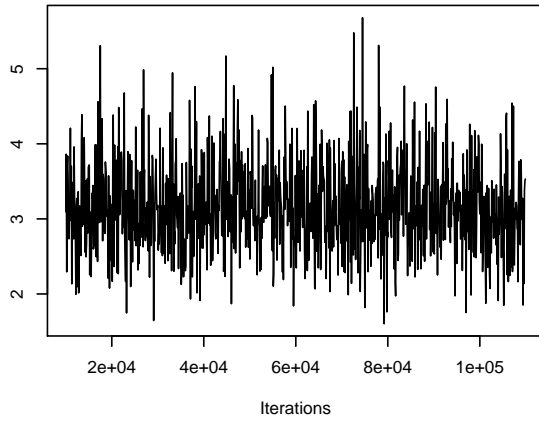
Trace of year



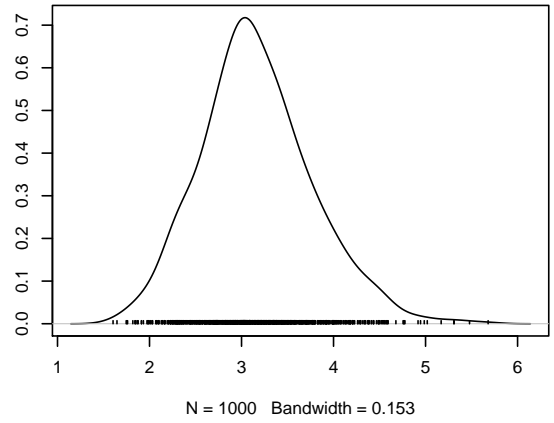
Density of year



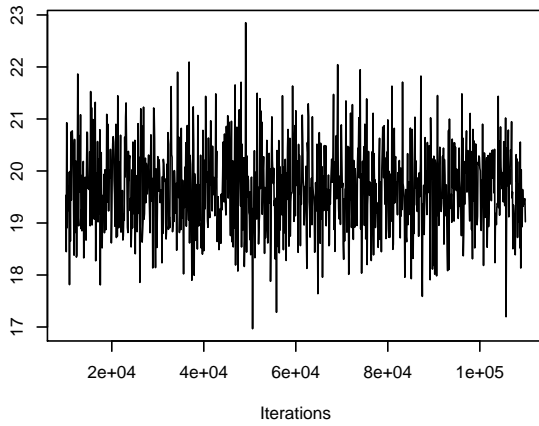
Trace of id



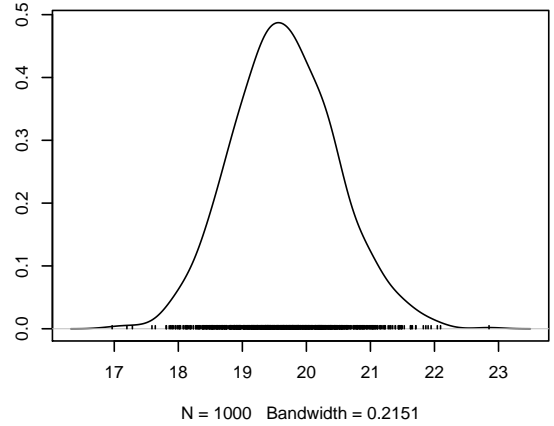
Density of id



Trace of units



Density of units



Random regression for FFD, including random effects of individual slopes

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

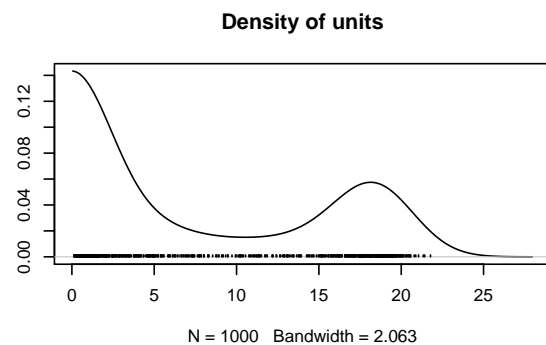
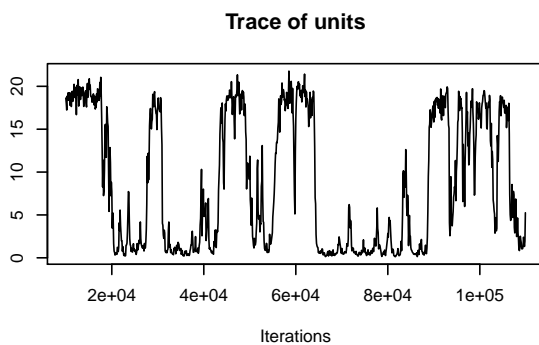
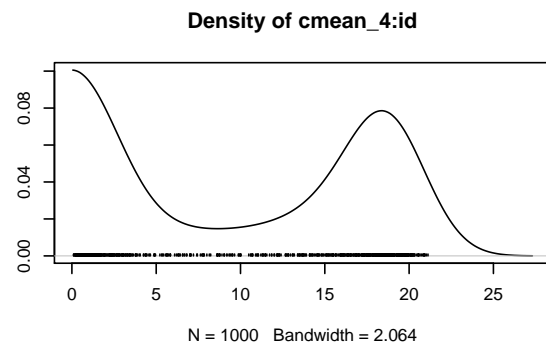
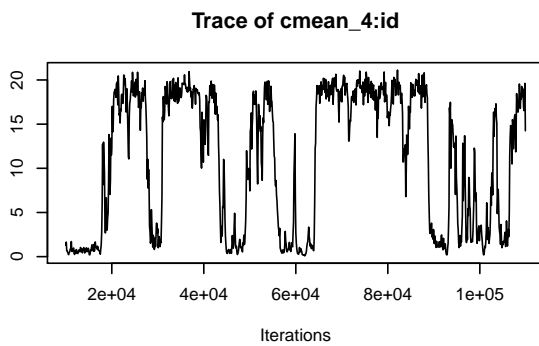
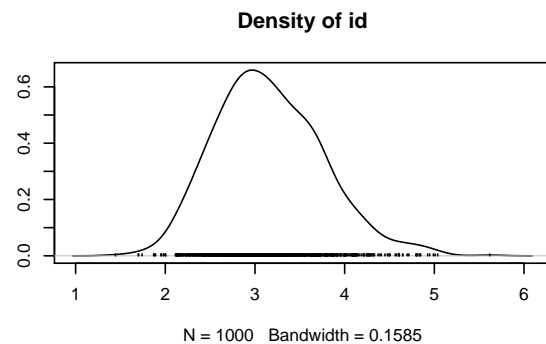
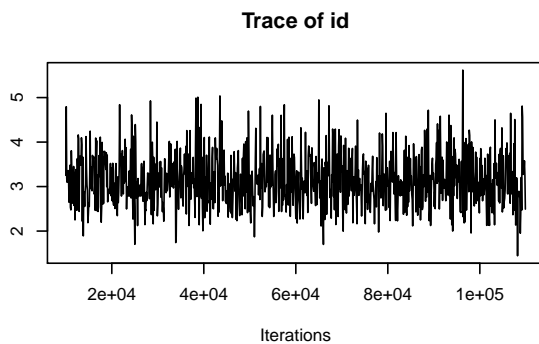
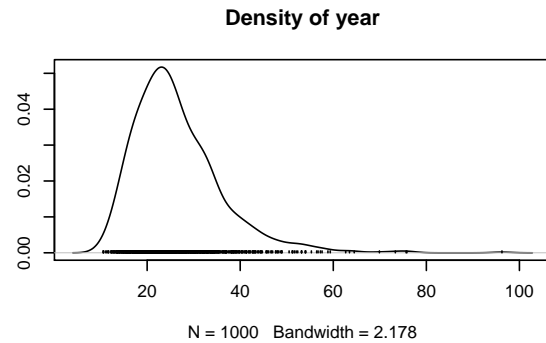
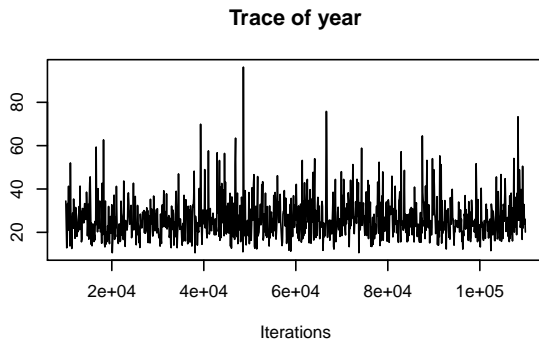
```
priorUV3_NoCov <- list(G = list(G1 = list(V = diag(1), nu = 1),  
                                # other random effect (YEAR)  
                                G2 = list(V = diag(1), nu = 1), # variance in intercepts  
                                G3 = list(V = diag(1), nu = 1)), # variance in 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: 5897.627  
##  
## G-structure: ~year  
##  
##      post.mean l-95% CI u-95% CI eff.samp  
## year      26.47      11.3      44.76      1000  
##  
##      ~id  
##  
##      post.mean l-95% CI u-95% CI eff.samp  
## id         3.158      2.116      4.293      1000  
##  
##      ~cmean_4:id  
##  
##      post.mean l-95% CI u-95% CI eff.samp  
## cmean_4:id     11.05      0.1893      19.84      13.27  
##  
## R-structure: ~units  
##  
##      post.mean l-95% CI u-95% CI eff.samp  
## units         8.629      0.2076      19.57      13.64  
##  
## Location effects: FFD ~ cmean_4  
##  
##      post.mean l-95% CI u-95% CI eff.samp pMCMC  
## (Intercept)   58.081   56.084   60.324     1000 <0.001 ***  
## cmean_4       -3.248   -5.541   -1.077     1151  0.002 **  
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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



Including covariance between intercept and slope

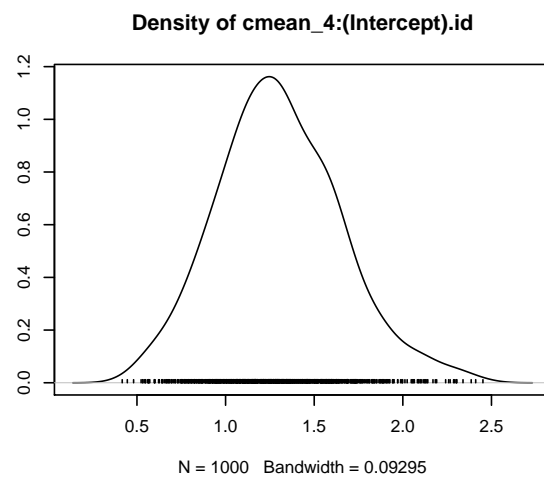
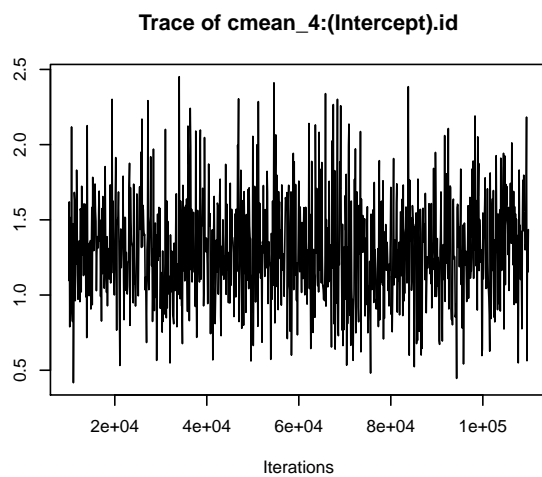
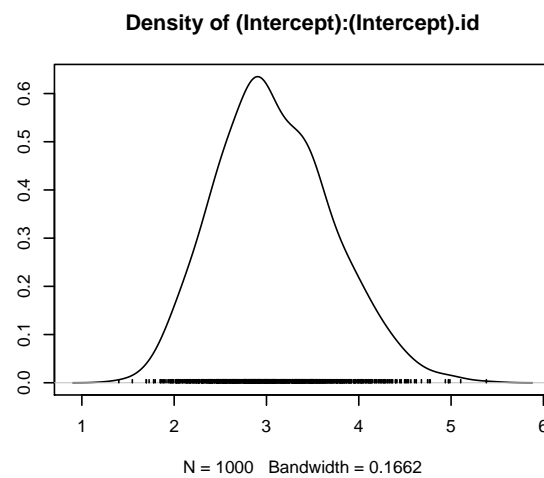
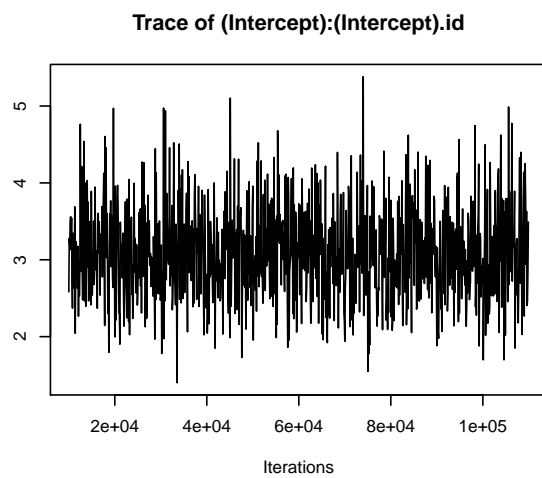
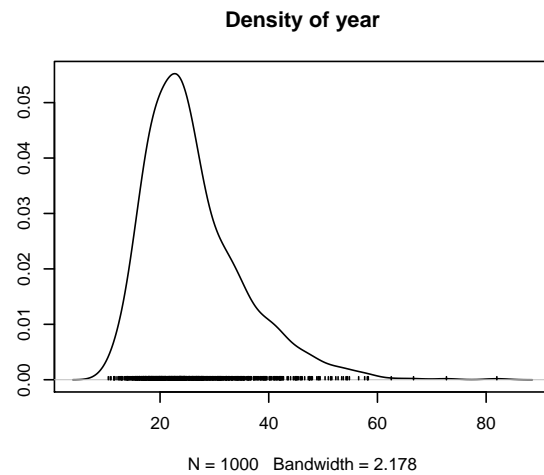
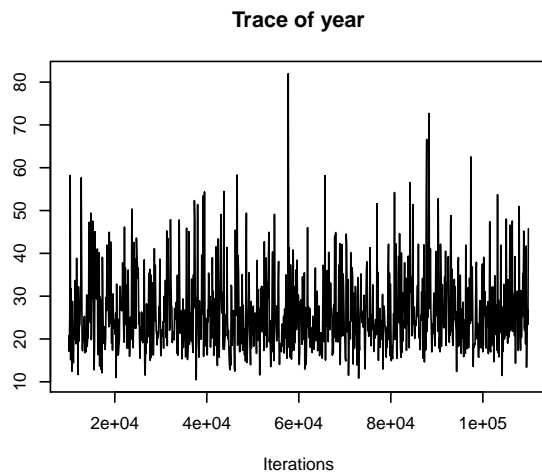
```
priorUV2_RR <- list(G = list(G1 = list(V = diag(1), nu = 1), # other random effect (YEAR)
                             G2 = list(V = diag(2), nu = 1)),
                   # ^ 2x2 variance-covariance matrix for var in slopes + intercepts
                   R = list(R1 = list(V = diag(1), nu = 2)))

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)

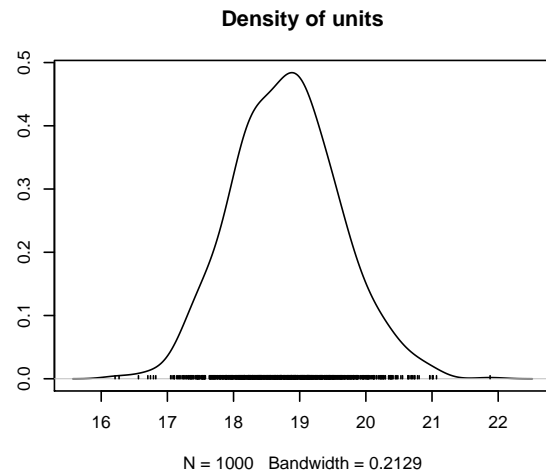
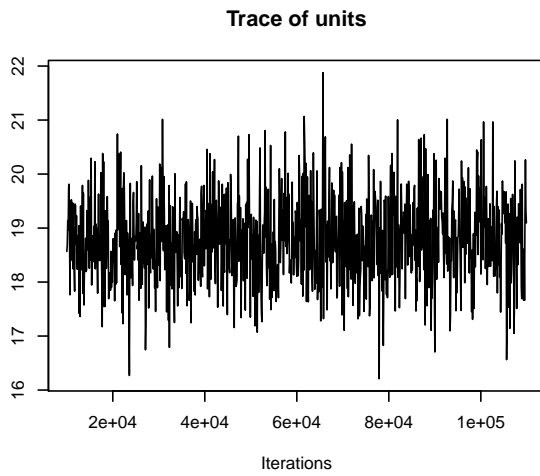
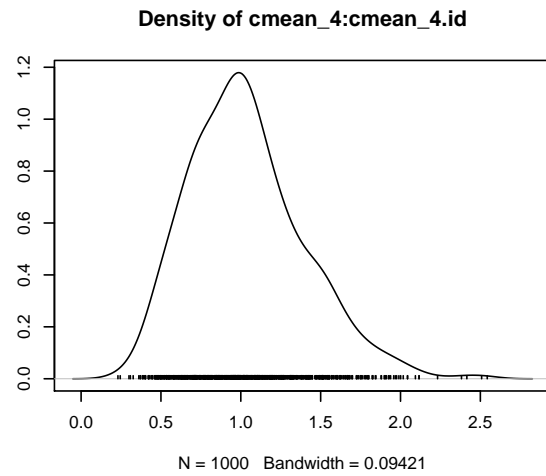
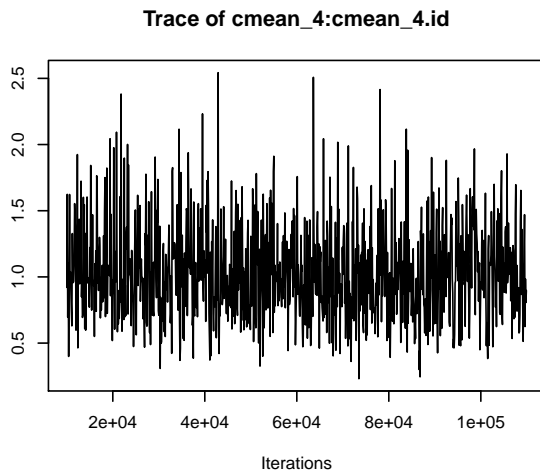
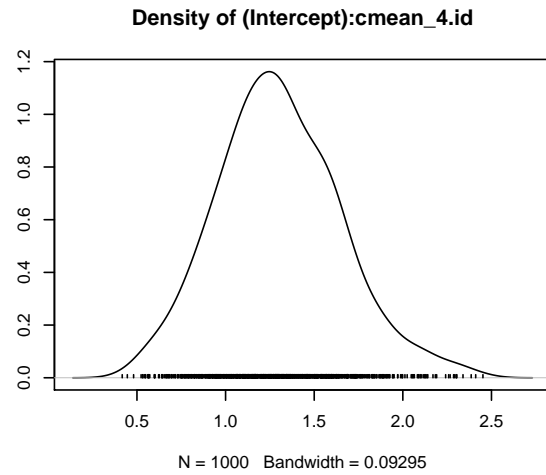
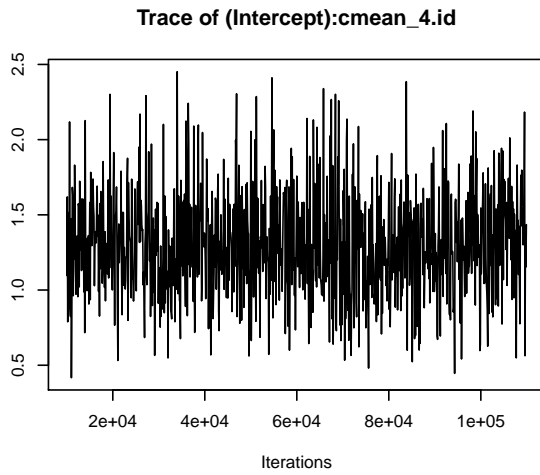
summary(univar.FFD_RR)
```

```
##
## Iterations = 10001:109901
## Thinning interval = 100
## Sample size = 1000
##
## DIC: 8564.645
##
## G-structure: ~year
##
##      post.mean l-95% CI u-95% CI eff.samp
## year      26.21   12.08   45.44     1000
##
##      ~us(1 + cmean_4):id
##
##      post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).id    3.086   1.9551   4.313   1209.4
## cmean_4:(Intercept).id        1.302   0.5613   1.950   998.5
## (Intercept):cmean_4.id        1.302   0.5613   1.950   998.5
## cmean_4:cmean_4.id            1.035   0.3992   1.758   1000.0
##
## R-structure: ~units
##
##      post.mean l-95% CI u-95% CI eff.samp
## units      18.79   17.31   20.39     1000
##
## Location effects: FFD ~ cmean_4
##
##      post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)   58.027   55.738   59.961     1000 <0.001 ***
## cmean_4       -3.189   -5.270   -1.096     1097 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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



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



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

Bivariate models of both fitness and FFD

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.

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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):obs,
  # ^ residual variance within individuals, between years
  # (labelled by 'Obs')
  data = data.stack,
  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	58.093	55.961	60.375	1096	0.001
variablefitness	1.000	0.878	1.129	1000	0.001
at.level(variable, "FFD"):temp	-3.211	-5.305	-1.033	1000	0.010

```
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	26.363	13.324	44.797	1145

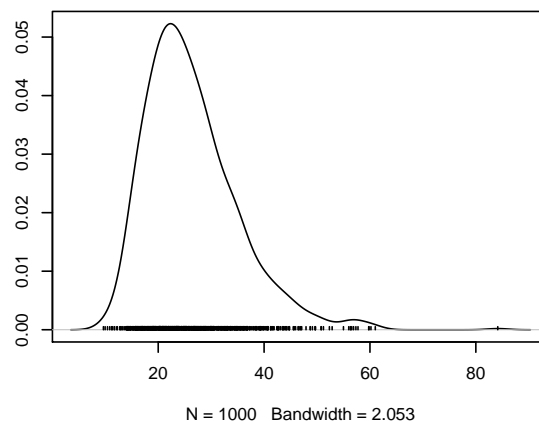
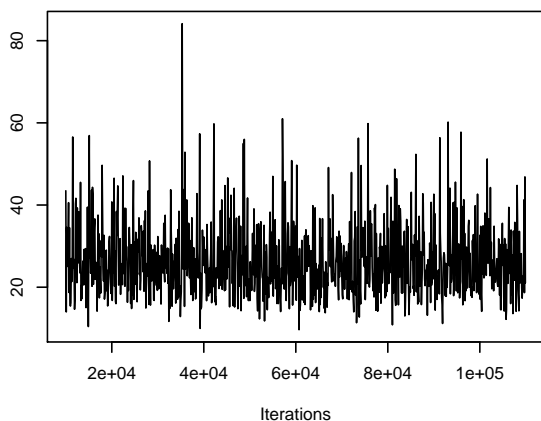
```
kable(summary(modelBV)$Rcovariances,digits=c(3,3,3,0),caption="Random effects")
```

Table 3: Random effects

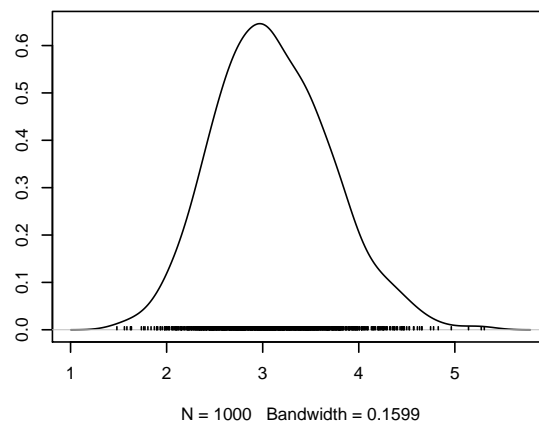
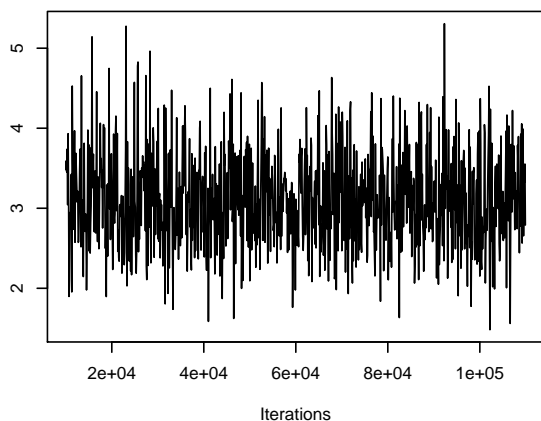
	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, "FFD").id:at.level(variable, "FFD").id	3.107	1.978	4.296	1141
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.604	-0.888	-0.291	1132
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.604	-0.888	-0.291	1132
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.966	0.790	1.137	1000
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	19.676	18.260	21.403	1000

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

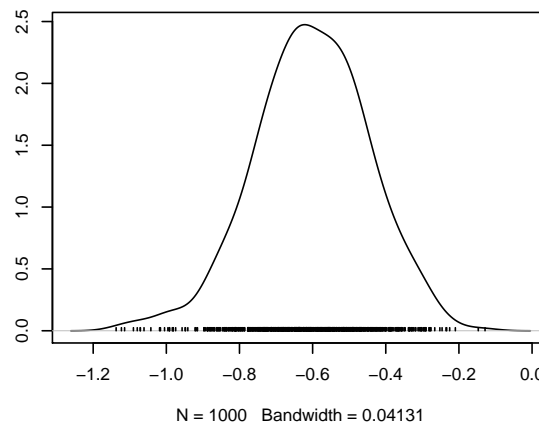
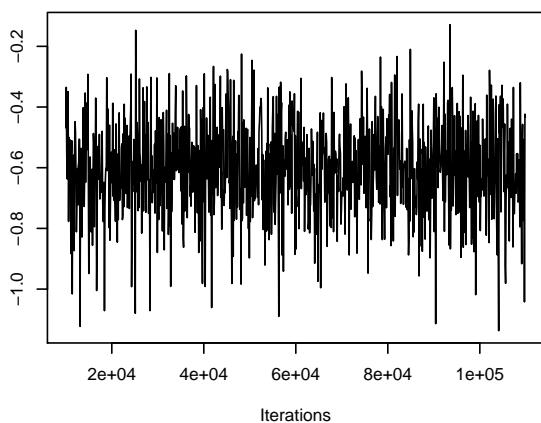
Trace of at.level(variable, "FFD"):at.level(variable, "FFD").y Density of at.level(variable, "FFD"):at.level(variable, "FFD").y



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

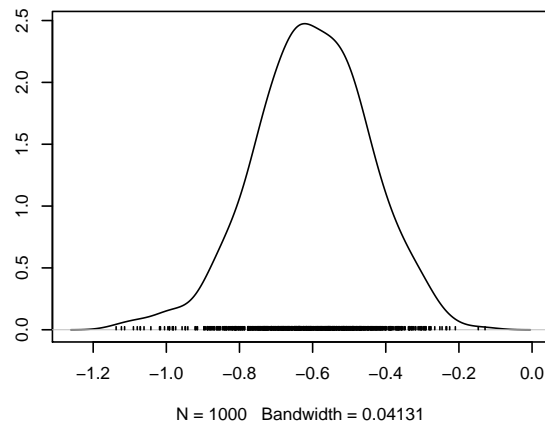
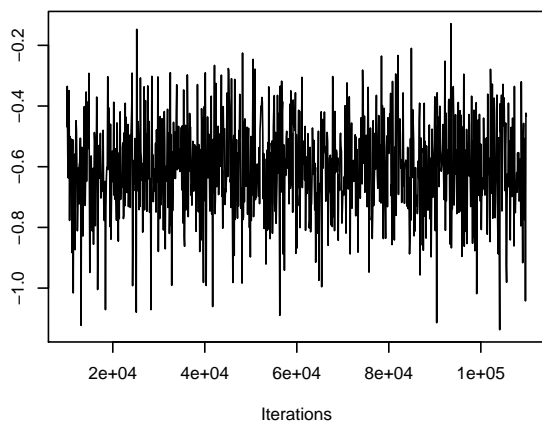


Trace of at.level(variable, "fitness").id:at.level(variable, "FFD") Density of at.level(variable, "fitness").id:at.level(variable, "FFD")

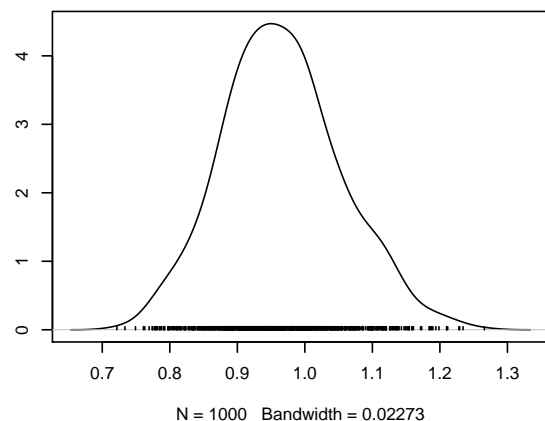
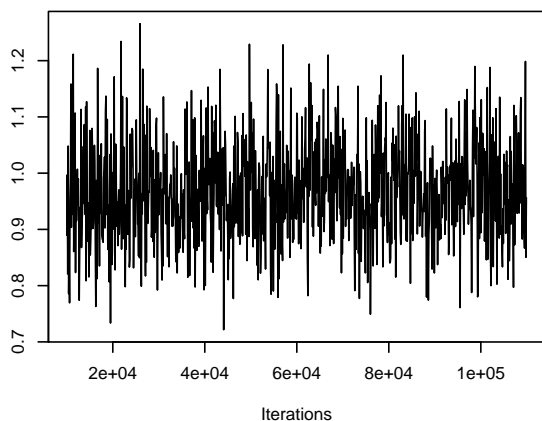


```
plot(modelBV$VCV[,4:6])
```

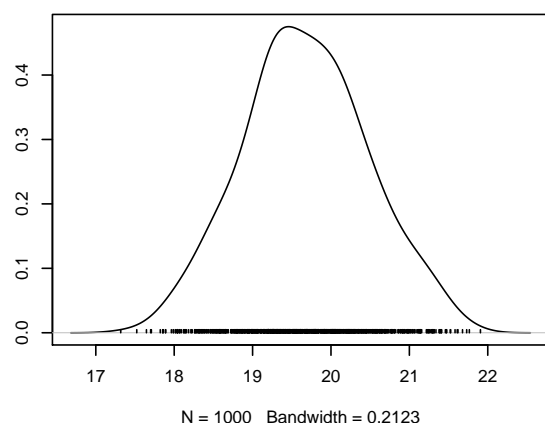
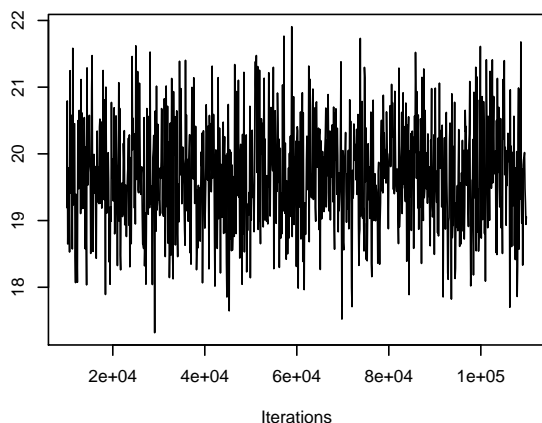
Trace of at.level(variable, "FFD").id:at.level(variable, "fitness")density of at.level(variable, "FFD").id:at.level(variable, "fitness")



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



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



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.


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: $\sim \text{us}(\text{at.level}(\text{trait}, 1)):\text{year} + \text{us}(\text{trait} + \text{cmean_4}:\text{at.level}(\text{trait}, 1)):\text{id}$ Having random intercepts for fitness means that there is variance in intercepts for relative fitness - don't think we need that - we consider the variance between individuals in fitness as residual variance. They also used $\text{rcov} = \sim \text{idh}(\text{trait}):\text{units}$. 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 individuals in fitness
  # (which is residual variance)
  us(at.level(variable, "FFD")):Obs,
  # ^ residual variance within individuals between years
  # (labelled by 'Obs')
  data = data.stack,
  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	58.108	56.014	60.489	1000	0.001
variablefitness	1.003	0.887	1.130	1000	0.001
at.level(variable, "FFD"):temp	-3.191	-5.031	-0.936	1000	0.006

```
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
 vel(variable, "FFD"):at.level(variable, "FFD").year	27.009	10.733	44.974	1000


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

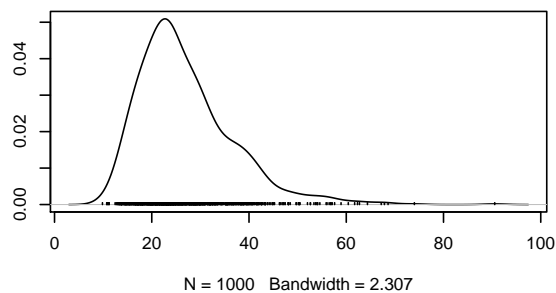
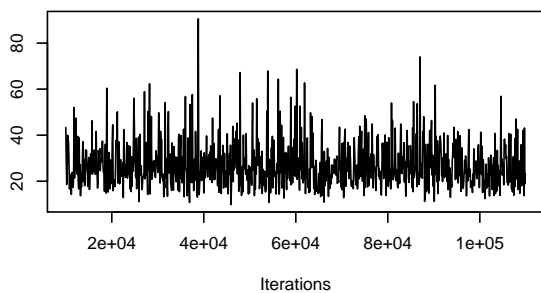
Table 6: Random effects

	post.mean	l-95% CI	u-95% CI	eff.samp
at.level(variable, "FFD").id:at.level(variable, "FFD").id	2.987	1.743	4.148	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD").id	1.186	0.616	1.904	1000
at.level(variable, "fitness").id:at.level(variable, "FFD").id	-0.581	-0.883	-0.269	888
at.level(variable, "FFD").id:at.level(variable, "FFD"):temp.id	1.186	0.616	1.904	1000
at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):temp.id	1.126	0.494	1.787	1000
at.level(variable, "fitness").id:at.level(variable, "FFD"):temp.id	-0.143	-0.397	0.140	1000
at.level(variable, "FFD").id:at.level(variable, "fitness").id	-0.581	-0.883	-0.269	888
at.level(variable, "FFD"):temp.id:at.level(variable, "fitness").id	-0.143	-0.397	0.140	1000
at.level(variable, "fitness").id:at.level(variable, "fitness").id	0.977	0.810	1.160	1000
at.level(variable, "FFD"):at.level(variable, "FFD").Obs	18.816	17.286	20.517	1000

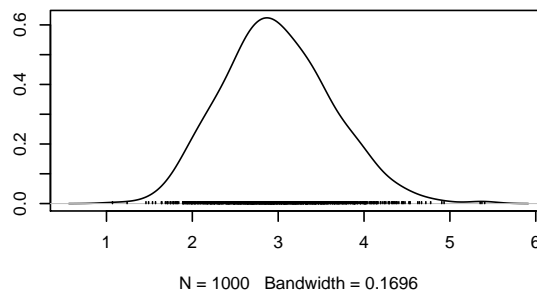
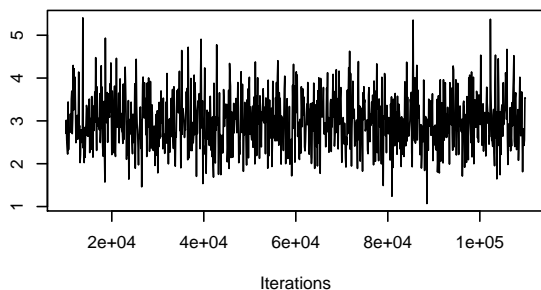


```
plot(modelBV_RR$VCV[,1:4])
```

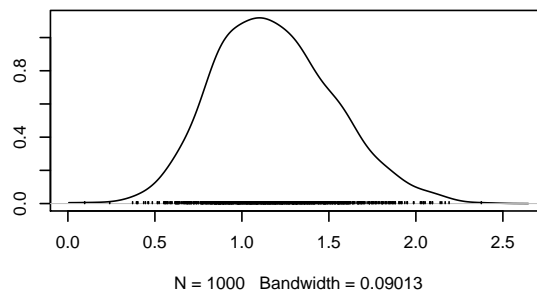
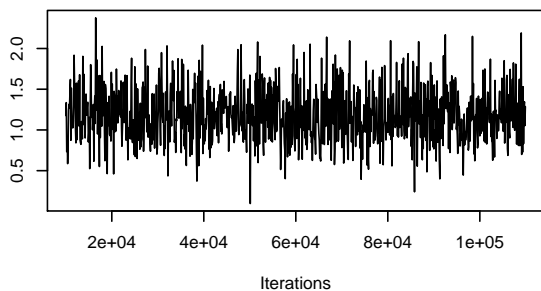
Trace of at.level(variable, "FFD"):at.level(variable, "FFD").ye Density of at.level(variable, "FFD"):at.level(variable, "FFD").y



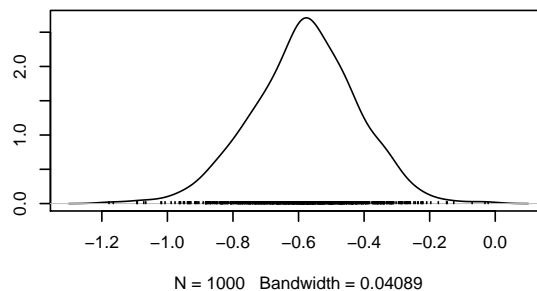
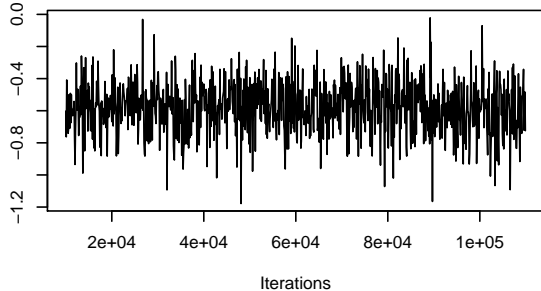
Trace of at.level(variable, "FFD").id:at.level(variable, "FFD") Density of at.level(variable, "FFD").id:at.level(variable, "FFD")



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

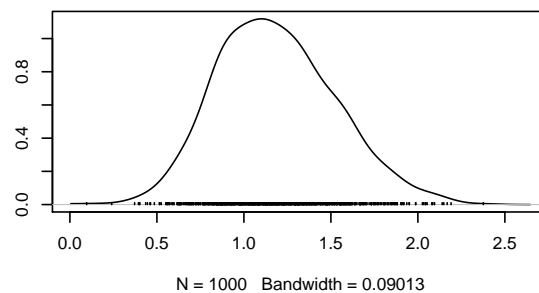
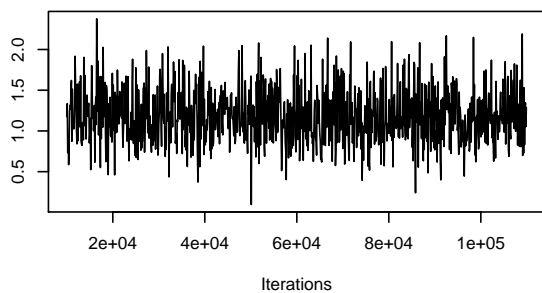


Trace of at.level(variable, "fitness").id:at.level(variable, "FFD") Density of at.level(variable, "fitness").id:at.level(variable, "FFD")

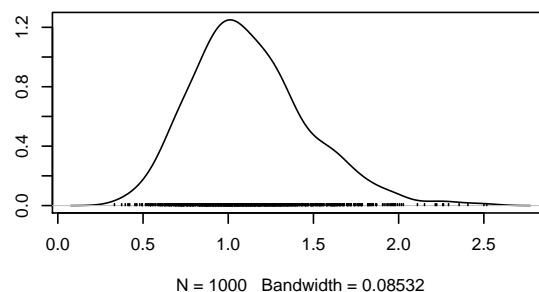
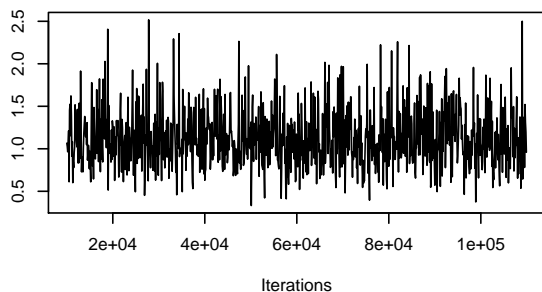


```
plot(modelBV_RR$VCV[,5:8])
```

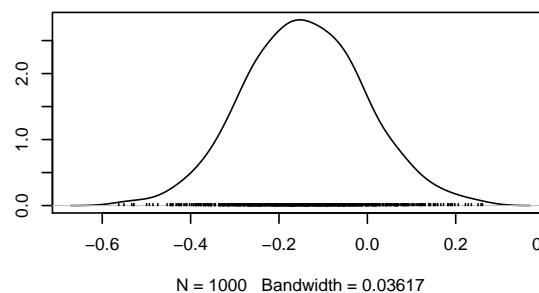
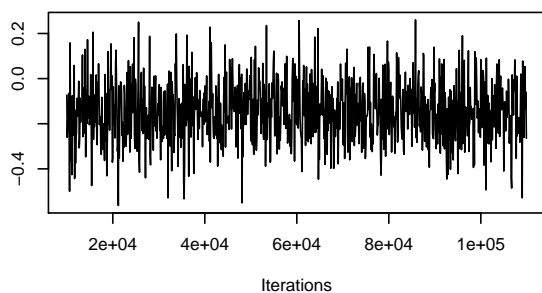
Trace of at.level(variable, "FFD").id:at.level(variable, "FFD"):tensity of at.level(variable, "FFD").id:at.level(variable, "FFD"):t



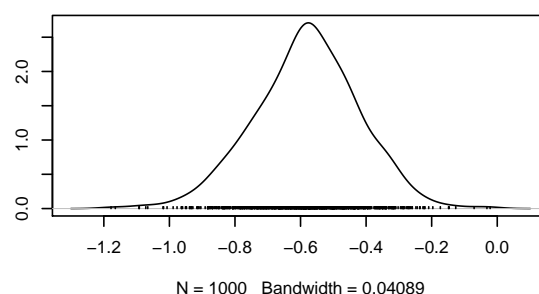
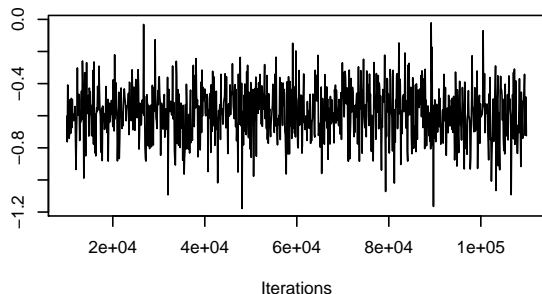
Trace of at.level(variable, "FFD"):temp.id:at.level(variable, "FFD"):ity of at.level(variable, "FFD"):temp.id:at.level(variable, "FFD")



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

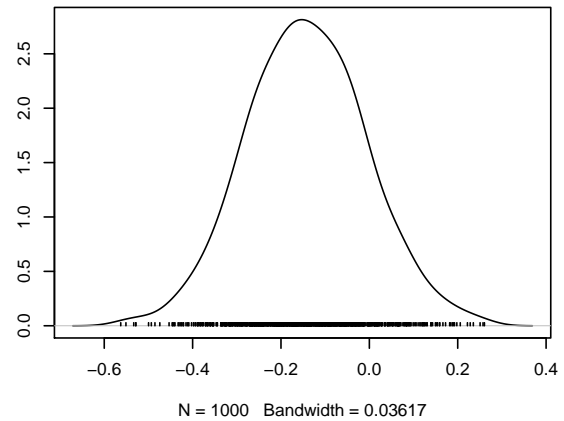
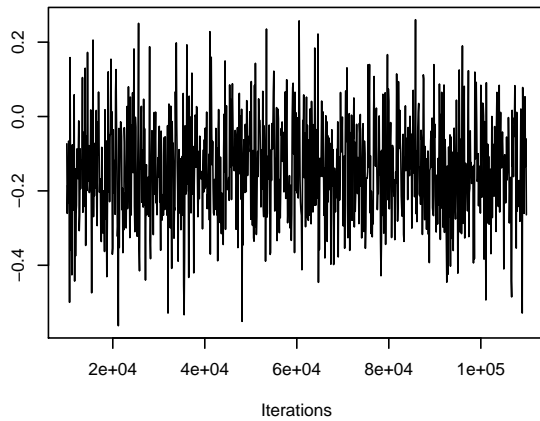


Trace of at.level(variable, "FFD").id:at.level(variable, "fitness"):ensity of at.level(variable, "FFD").id:at.level(variable, "fitness")

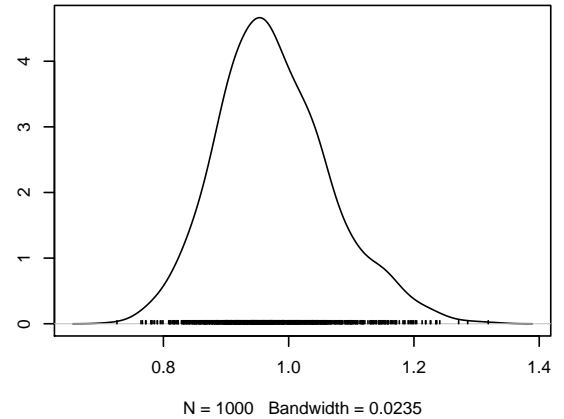
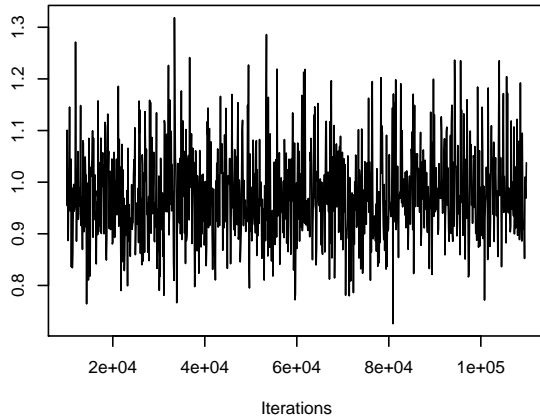


```
plot(modelBV_RR$VCV[,9:11])
```

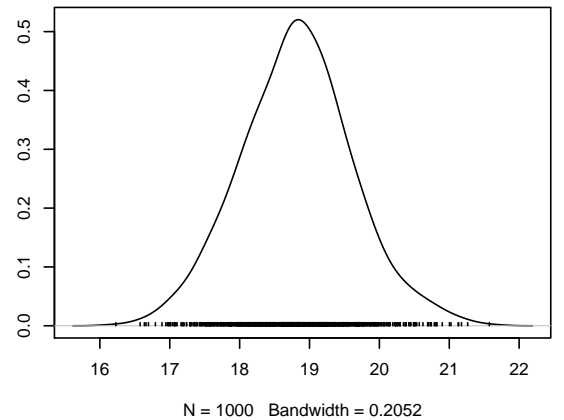
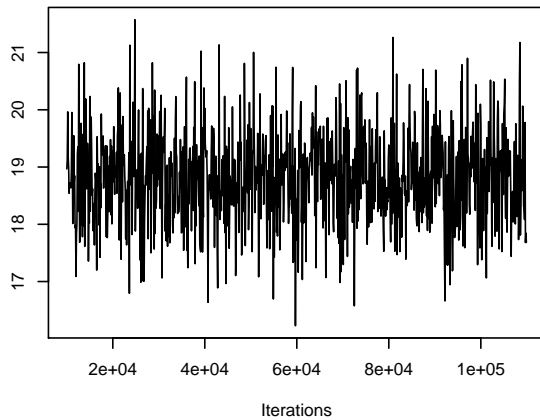
Trace of at.level(variable, "FFD"):temp.id:at.level(variable, "fitness"):temp.id:at.level(variable, "fitn



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



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



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,]  2.9217972  0.9837027 -0.5871924
## [2,]  0.9837027  1.0124701 -0.1700363
## [3,] -0.5871924 -0.1700363  0.9435179

# 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] -0.5871924 -0.1700363

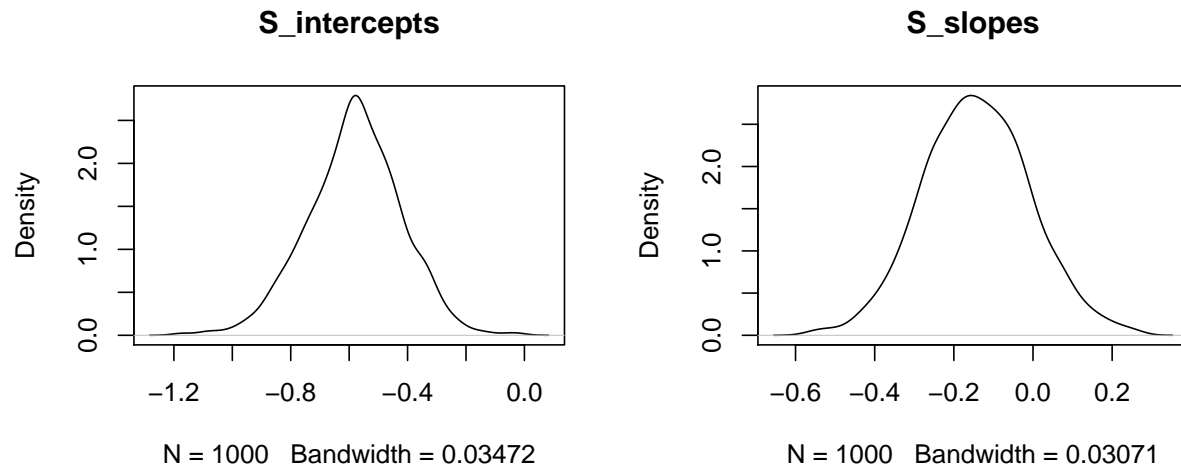
posterior.mode(mcmc(S.modelBV_RR))

## S_intercepts    S_slopes
##    -0.5871924    -0.1700363

HPDinterval(mcmc(S.modelBV_RR))

##           lower      upper
## S_intercepts -0.8833353 -0.2694784
## S_slopes     -0.3971602  0.1398005
## attr(,"Probability")
## [1] 0.95

# 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")
```



```
# 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
##      -0.23750071      0.09846645

HPDinterval(mcmc(beta_post_RR))

##                lower      upper
## beta_intercepts -0.4847463 -0.03976347
## beta_slopes     -0.2230782  0.60421496
## 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")
```

