

# Lv 15 Heritability, Gen. sel

## Prep

Load packages

```
rm(list=ls())
library(lattice)
library(lme4)
```

## Loading required package: Matrix

```
library(ggplot2)
source("HighstatLibV10.R")
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      0.00   5.00   14.00   24.77   31.00   214.68       98

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00   1.00   10.00   20.55   26.40   214.68

## 'data.frame':   575 obs. of  12 variables:
## $ Mum      : Factor w/ 54 levels "10:3:1","11:5:3",...: 45 18 18 18 18 18 18 18 18 ...
## $ Dad      : Factor w/ 64 levels "10:3:1","11:1:2",...: 15 39 39 39 39 39 39 61 61 61 ...
## $ PlantID   : Factor w/ 575 levels " 497:3","100:1",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ volume    : num  2347 1755 1122 8750 2053 ...
## $ FFD       : num  140 128 128 117 123 ...
## $ allseeds  : num  3 NA 14 5 21 ...
## $ kod_grazing: Factor w/ 2 levels "grazed","OK": 2 1 2 1 1 1 1 1 1 2 ...
## $ index_all : Factor w/ 2 levels "est","obs": 2 1 2 1 1 1 1 1 1 1 ...
## $ std.volume : num  -0.236 -0.505 -0.794 2.681 -0.369 ...
## $ std.FFD    : num  2.44 0.445 0.311 -1.446 -0.5 ...
## $ N.seeds    : num  3 0 14 5 21 ...
## $ rel.seeds  : num  0.146 0 0.681 0.243 1.022 ...

##      Mum      Dad      PlantID      volume
## 39:5:2 : 37   8:3:2 : 28   497:3 : 1   Min.    : 242.2
## 32:5:1 : 32   8:5:1 : 26   100:1 : 1   1st Qu.: 1313.4
## 2:1:3  : 27   31:1:2 : 25   100:2 : 1   Median : 2263.7
## 12:8:2 : 24   21:4:1 : 20   100:3 : 1   Mean    : 2864.3
## 19:1:4 : 22   9:6:2  : 20   100:6 : 1   3rd Qu.: 3798.6
## 15:3:1 : 21   16:1:2 : 19   100:7 : 1   Max.    :13730.5
## (Other):412 (Other):437 (Other):569

##      FFD      allseeds      kod_grazing      index_all      std.volume
## Min.    :112.0   Min.    : 0.00   grazed:316   est:387   Min.    :-1.1942
## 1st Qu.:121.5   1st Qu.: 5.00   OK      :259   obs:188   1st Qu.: -0.7063
## Median :126.0   Median : 14.00                      Median :-0.2735
## Mean    :125.7   Mean    : 24.77                      Mean    : 0.0000
## 3rd Qu.:128.3   3rd Qu.: 31.00                      3rd Qu.: 0.4255
## Max.    :143.5   Max.    :214.68                      Max.    : 4.9489
##
##      NA's :98

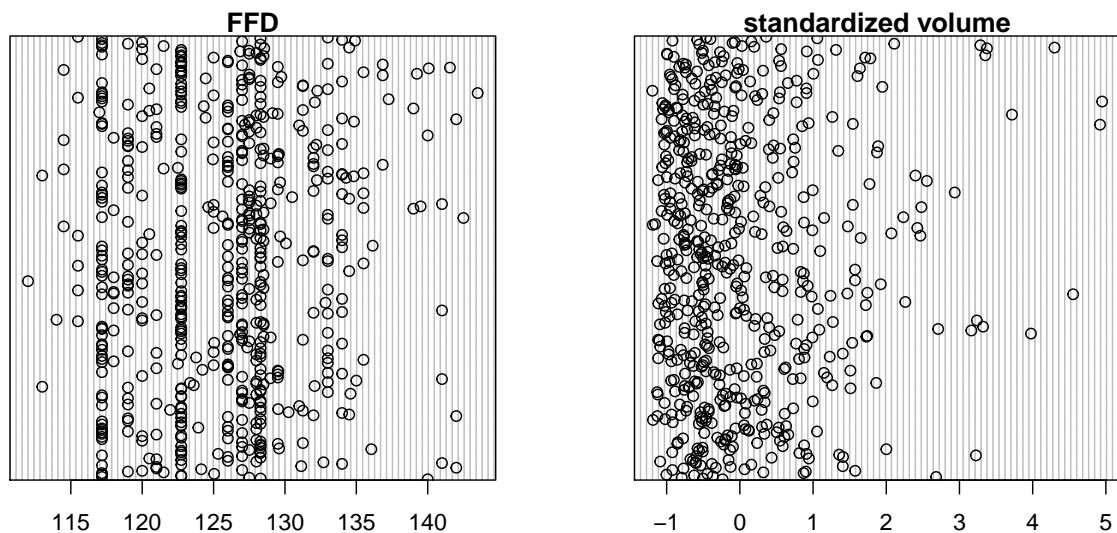
##      std.FFD      N.seeds      rel.seeds
## Min.    :-2.32872   Min.    : 0.00   Min.    : 0.00000
## 1st Qu.: -0.71077   1st Qu.: 1.00   1st Qu.: 0.04866
```

```
## Median : 0.05563   Median : 10.00   Median : 0.48657
## Mean   : 0.00000   Mean   : 20.55   Mean   : 1.00000
## 3rd Qu.: 0.44536   3rd Qu.: 26.40   3rd Qu.: 1.28456
## Max.   : 3.03608   Max.    :214.68   Max.    :10.44599
##
```

## HERITABILITY

### Data exploration

```
# OUTLIERS?
par(mfrow = c(1,2), mar = c(3, 4, 1, 1))
dotchart(S2$FFD, main= "FFD") #OK
#dotchart(S2$volume, main= "volume") #yes
dotchart(S2$std.volume, main= "standardized volume") #better
```



Data is not balanced:

```
#Balanced data?
table(S2$Dad) #
```

```
##
##      10:3:1      11:1:2      11:3:1      11:5:3      11:6:1
##           3          16           8           5           2
##      11:8:1      12:10:2     12:3:1      12:7:5      12:8:2
##          11           6           5           3           6
##      12:9:2      13:6:1      13:8:1      13:8:4      14:2:2
##           6           9           6          12          11
##      14:5:5      15:3:2      16:1:1      16:1:2      16:2:1
##          13           6           5          19           2
##      18:4:2      19:1:7      19:3:4      19:6:2      19:6:5
##           3           7          14           9           2
##       2:4:1      21:1:2      21:4:1      22:4:1      23:1:2
##           5           2          20          12           4
##      23:1:4      24:2:1      24:3:2      24:4:4      24:5:1
```

```
##          5          6          7          4          5
##      27:2:1      27:4:4      3:8:4      31:1:2      31:3:4
##          5          10          10          25          8
##      31:3:5      31:4:6      32:2:2      35:2:2      39:2:3
##          11          4          6          7          15
##      39:3:3      39:4:2      39:5:2      39:5:3      4:1:4
##          19          7          5          2          16
##      5:1:1      6:1:2      6:2:3      6:4:2      6:5:1
##          4          10          12          8          6
##      7:1:1      8:3:1      8:3:2      8:4:1      8:5:1
##          8          7          28          8          26
##      9:1:6      9:6:2      9:7:3 98_30_1:4:1 T2
##          8          20          13          8
```

```
range(table(S2$Dad))
```

```
## [1]  2 28
```

```
table(S2$Mum)
```

```
##
##      10:3:1      11:5:3      11:6:3      11:8:1
##          9          2          2          2
##      12:10:2      12:3:1      12:3:4      12:4:5
##          7          7          13          12
##      12:4:6      12:6:1      12:7:5      12:8:2
##          14          9          3          24
##      12:9:2 129_25_2:4:2 T2      13:6:2      13:7:1
##          16          10          10          9
##      13:8:4      14:2:2      15:3:1      15:3:2
##          2          13          21          5
##      16:1:1      16:5:1      19:1:4      19:1:7
##          9          10          22          3
##      19:2:4      19:4:2      19:5:1      19:5:2
##          2          3          7          7
##      19:5:5      19:6:5      2:1:3      2:4:1
##          19          9          27          12
##      21:2:4      21:3:3      21:5:3      22:4:1
##          7          2          8          15
##      24:1:1      32:2:2      32:5:1      32:8:3
##          2          3          32          21
##      33:1:1      36:3:3      38:2:1      39:2:2
##          6          8          14          5
##      39:3:2      39:3:3      39:4:2      39:5:2
##          8          11          15          37
##      39:5:3      4:1:6      6:1:1      6:2:3
##          9          8          15          15
##      8:1:2      9:7:3
##          4          10
```

```
range(table(S2$Mum))
```

```
## [1]  2 37
```

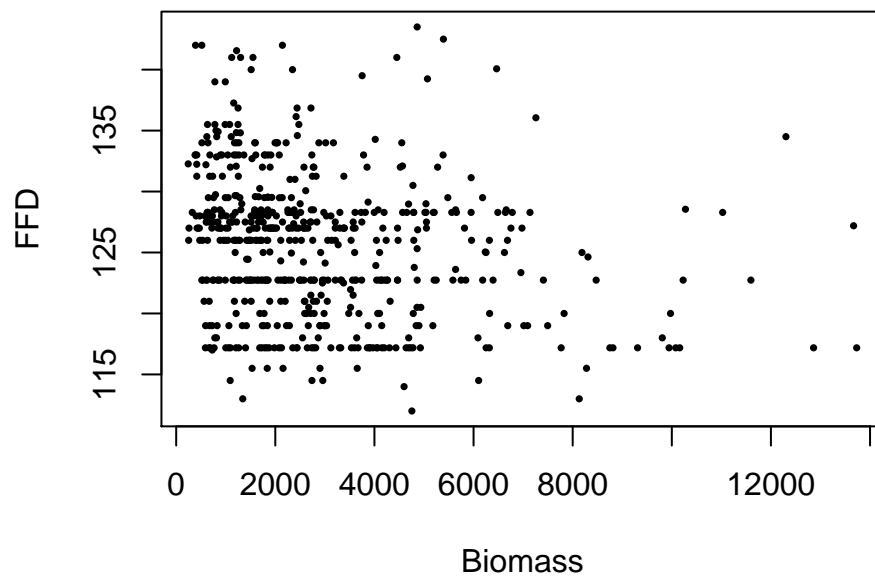
```
# RELATIONSHIPS?
```

```
plot(x = S2$volume,
```

```

y = S2$FFD,
pch = 16, cex = 0.5,
ylab = "FFD",
xlab = "Biomass")

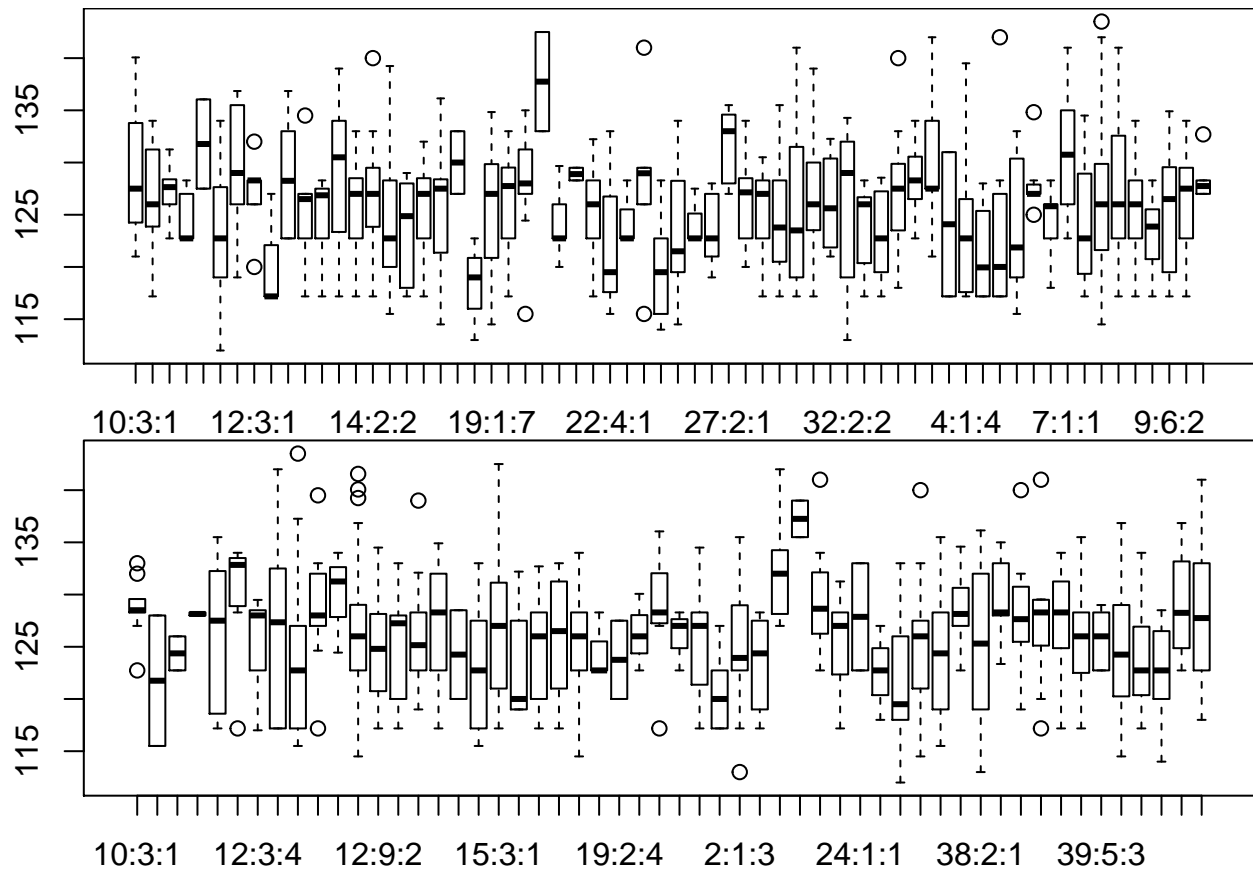
```



```

#And the random effects:
par(mfrow=c(2,1), mar=c(2,2,0,0))
boxplot(FFD ~ Dad, data = S2)
boxplot(FFD ~ Mum, data = S2)

```

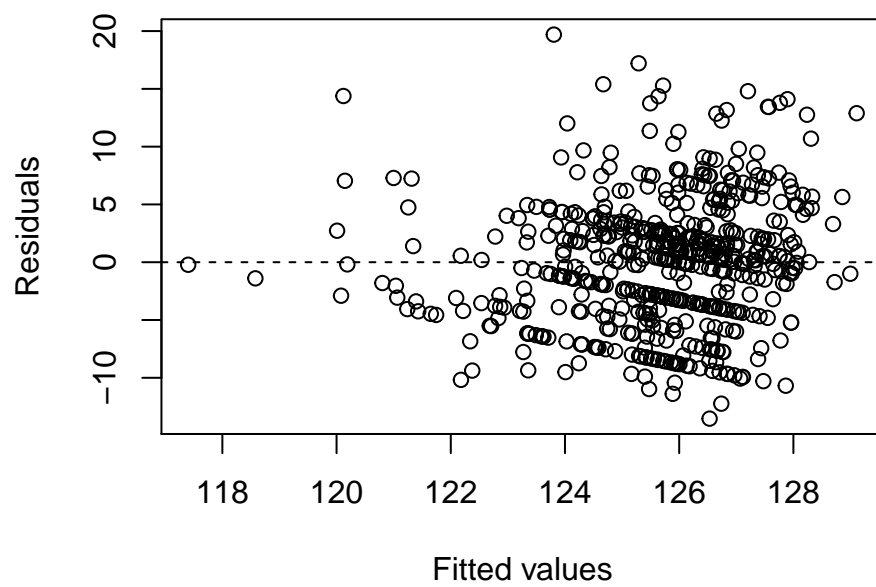


## Construct model

```
#Step 1 of the protocol
modh2 <- lmer(FFD ~ std.volume + (1 | Dad) + (1 | Mum),
  data = S2,
  REML = TRUE)
```

```
E1 <- resid(modh2)
F1 <- fitted(modh2)

par(mfrow = c(1, 1))
plot(x = F1,
  y = E1,
  xlab = "Fitted values",
  ylab = "Residuals")
abline(v = 0, lwd = 2, col = 2)
abline(h = 0, lty = 2, col = 1)
```

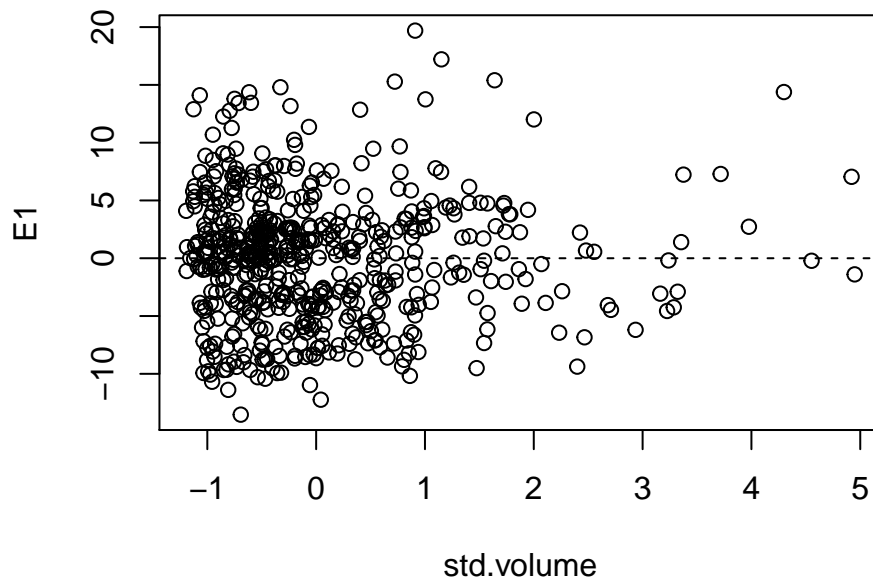


*#look for fitted values outside of possible obs vals*

*#1.2*

```
plot(E1 ~ std.volume, data = S2)  
abline(h = 0, lty = 2)
```

-673.696		logL_red	LR_D	LRT_P	LRT_df
	Mum	-675.5076	3.6232382	0.0284889	1
	Dad	-673.7404	0.0888847	0.3827999	1



```
#resids for std.volume=0 problem?
```

```
library(rptR)
```

```
#Likelihood ratio
```

```
#rep <- rpt(FFD ~ std.volume + (1 | Mum) + (1 | Dad), grname = c("Mum",  
#   "Dad"), data = S2, datatype = "Gaussian", nboot = 5000,  
#   npermut = 0)  
#save(rep, file="ICC_Boot.RData")
```

```
load("ICC_Boot.RData")
```

```
#str(rep)
```

```
knitr::kable(rep$LRT)
```

```
summary(rep)
```

```
##
```

```
## Repeatability estimation using the lmm method
```

```
##
```

```
## Call = rpt(formula = FFD ~ 1.flowers + (1 | Mum) + (1 | Dad), grname = c("Mum", "Dad"), data = S2, datatype =
```

```
##
```

```
## Data: 224 observations
```

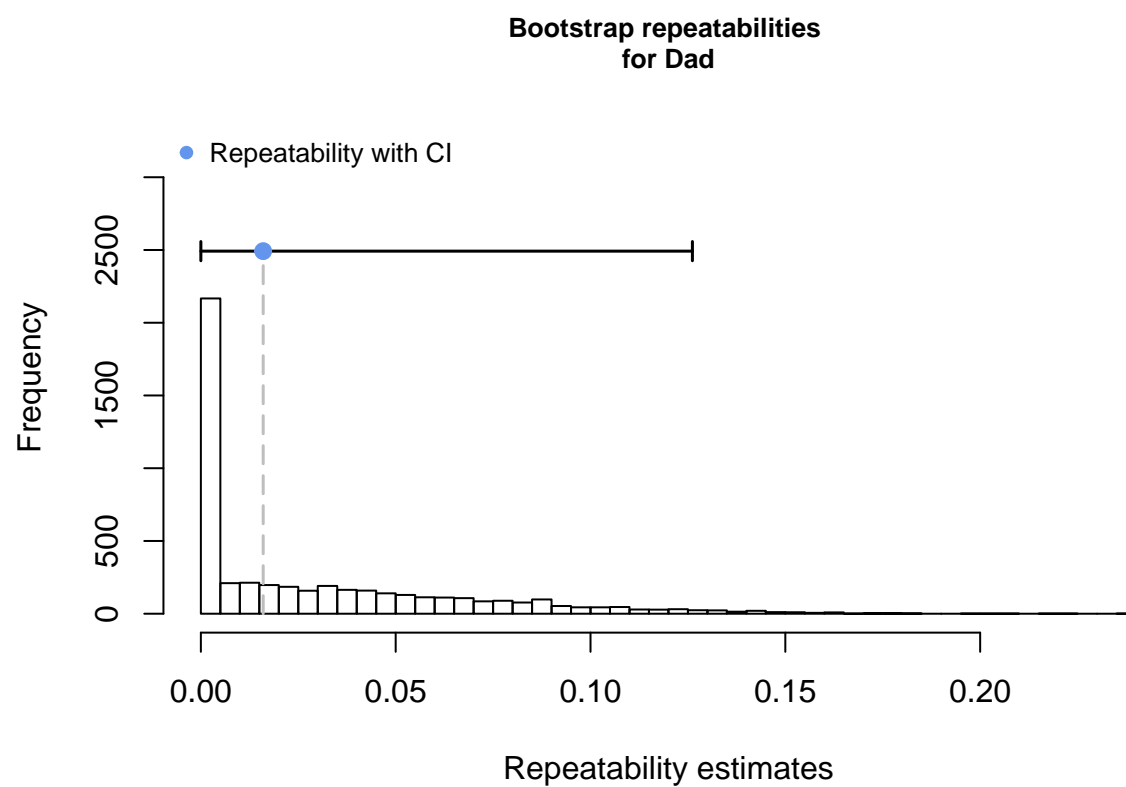
```
## -----
```

```

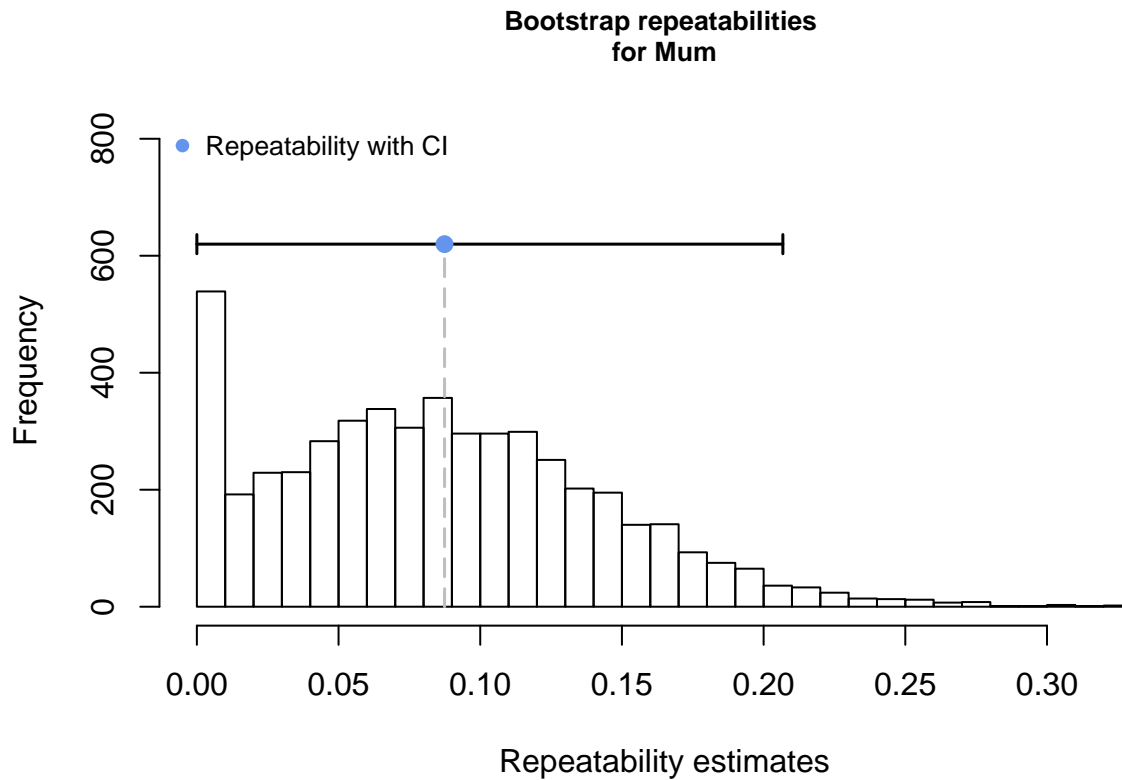
##
## Mum (41 groups)
##
## Repeatability estimation overview:
##      R      SE    2.5% 97.5% P_permut  LRT_P
## 0.0874 0.0571      0 0.207      NA 0.028
##
## Bootstrapping and Permutation test:
##      N    Mean Median    2.5% 97.5%
## boot   5000 0.0853 0.0816      0 0.207
## permut    1    NA     NA     NA    NA
##
## Likelihood ratio test:
## logLik full model = -673.696
## logLik red. model = -675.5076
## D = 3.62, df = 1, P = 0.0285
##
## -----
##
## Dad (46 groups)
##
## Repeatability estimation overview:
##      R      SE    2.5% 97.5% P_permut  LRT_P
## 0.016 0.0373      0 0.126      NA 0.383
##
## Bootstrapping and Permutation test:
##      N    Mean Median    2.5% 97.5%
## boot   5000 0.0292 0.013      0 0.126
## permut    1    NA     NA     NA    NA
##
## Likelihood ratio test:
## logLik full model = -673.696
## logLik red. model = -673.7404
## D = 0.0889, df = 1, P = 0.383
##
## -----
plot(rep, grname = "Dad", type = "boot", cex.main = 0.8)

```





```
plot(rep, grname = "Mum", type = "boot", cex.main = 0.8)
```



```
#permutation

#rep2 <- rpt(FFD ~ std.volume + (1 | Mum) + (1 | Dad), grname = c("Mum",
# "Dad"), data = S2, datatype = "Gaussian", nboot = 0,
# npermut = 5000)
#save(rep2, file="ICC_Permut.RData")
load("ICC_Permut.RData")
print(rep2)
```

```
##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Mum
## R = 0.087
## SE = NA
## CI = [NA, NA]
## P = 0.0285 [LRT]
## 0.0244 [Permutation]
##
## Repeatability for Dad
## R = 0.016
## SE = NA
## CI = [NA, NA]
## P = 0.383 [LRT]
## 0.345 [Permutation]
```

```

#LIKELIHOOD RATIO TEST:
modh2_LR.all <- lmer(FFD ~ std.volume + (1 | Mum) + (1 | Dad),
  data = S2,
  REML = FALSE) #OBS!!
#REML=FALSE because we want to compare models on likelihood

#Effect of volume?
drop1(modh2_LR.all, test = "Chi")

## Single term deletions
##
## Model:
## FFD ~ std.volume + (1 | Mum) + (1 | Dad)
##           Df      AIC      LRT Pr(Chi)
## <none>           3637.5
## std.volume  1 3664.2 28.657 8.64e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#modell utan random(dad)
modh2_LR.Mum = lmer(FFD ~ std.volume + (1 | Mum),
  data = S2,
  REML = FALSE)

#modell utan random(mum)
modh2_LR.Dad = lmer(FFD ~ std.volume + (1 | Dad),
  data = S2,
  REML = FALSE)

#Donor effect?
anova(modh2_LR.Mum, modh2_LR.all)

## Data: S2
## Models:
## modh2_LR.Mum: FFD ~ std.volume + (1 | Mum)
## modh2_LR.all: FFD ~ std.volume + (1 | Mum) + (1 | Dad)
##           Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## modh2_LR.Mum  4 3635.5 3653.0 -1813.8   3627.5
## modh2_LR.all  5 3637.5 3659.3 -1813.8   3627.5      0      1      1

#Recipient effect?
anova(modh2_LR.Dad, modh2_LR.all)

## Data: S2
## Models:
## modh2_LR.Dad: FFD ~ std.volume + (1 | Dad)
## modh2_LR.all: FFD ~ std.volume + (1 | Mum) + (1 | Dad)
##           Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## modh2_LR.Dad  4 3644.5 3661.9 -1818.3   3636.5
## modh2_LR.all  5 3637.5 3659.3 -1813.8   3627.5 8.9799      1 0.00273 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#####
#Step 3: Presents results (REML gives better estimates of the sigmas...
#           so use REML for final presentation).

```

```

#           So that is modh2

#Understand the output:
summary(modh2)

## Linear mixed model fit by REML ['lmerMod']
## Formula: FFD ~ std.volume + (1 | Dad) + (1 | Mum)
## Data: S2
##
## REML criterion at convergence: 3629
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.4369 -0.6985  0.0495  0.5296  3.5465
##
## Random effects:
## Groups Name Variance Std.Dev.
## Dad (Intercept) 1.501e-14 1.225e-07
## Mum (Intercept) 1.988e+00 1.410e+00
## Residual 3.084e+01 5.553e+00
## Number of obs: 575, groups: Dad, 64; Mum, 54
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 125.8628 0.3182 395.5
## std.volume -1.2940 0.2392 -5.4
##
## Correlation of Fixed Effects:
## (Intr)
## std.volume 0.019

```

## Variance Components

```

Variance <- as.data.frame(VarCorr(modh2))[,4] #extract variance components

#####
# Intra-class correlation
#####

# Proportional variance
# The correlation of two measurements in the same class

#https://stats.stackexchange.com/questions/113577/interpreting-the-random-effect-in-a-mixed-effect-mode

PropVar <- Variance/sum(Variance) #Proportional variance

## sigma_group1^2 /sum(sigma_group1^2, ... , sigma_resid^2)
# Same as:
## var_group1 /sum(var_group1, ... , var_resid)

#ICC <- (Variance)/sum(Variance)
#ICC[3] <- NA

```

```

# (hallsson och bjl. olika pga avrundade värden i tabell)

#####
# Likelihood ratio test var.comps
#####

#plocka ut chi^2 och p-värde från parvisa test
Chi2 <- c(anova(modh2_LR.Mum, modh2_LR.all)$Chisq[2],
          anova(modh2_LR.Dad, modh2_LR.all)$Chisq[2],
          NA)

Pvals <- c(anova(modh2_LR.Mum, modh2_LR.all)$"Pr(>Chisq)"[2],
          anova(modh2_LR.Dad, modh2_LR.all)$"Pr(>Chisq)"[2],
          NA)

#anova(modh2_LR.Dad, modh2_all)$"Chi Df"[2],

```

## Heritability & MaternalEffects

```

#####
# h^2 (paternal effects)
#####

## s^2_dad = 1/4 VA
## h2 = 4*s^2_dad / sum(s^2_dad,s^2_mum,s^2_resid)

#h2 <- (4 * Var_dad) /sum(Variance)
#h2 <- (4 * Variance[1]) /sum(Variance)

#same as

h2 <- c(4*PropVar[1], NA, NA)

#####
# maternal effects
#####

# Maternal - paternal
# -----
# summed effects

ME <- (Variance[2]-Variance[1])/sum(Variance)
ME <- c(NA, ME, NA)

options(knitr.kable.NA = '')

VC <- as.data.frame(cbind(Variance, PropVar, Chi2, Pvals, h2, ME))
colnames(VC)<- c("Variance <br/> component", "Proportional <br/> variance",
               "\u03C7^2~1~", "p-value",
               "h^2", "Maternal <br/> effect")

```

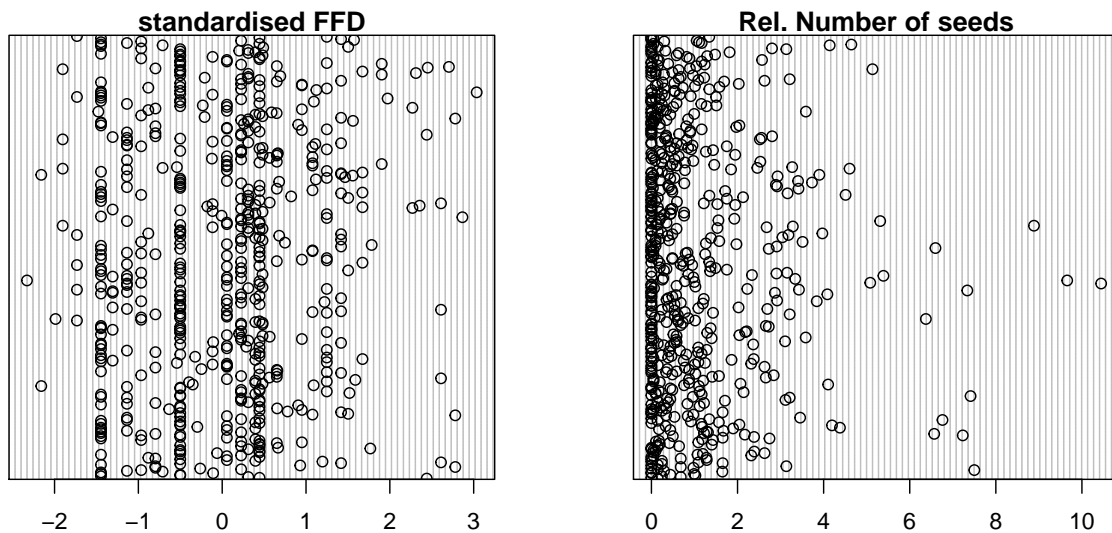
```
rownames(VC) <- c("Donor", "Recipient", "Residual")
knitr::kable(VC, digits=3, align = "l")
```

	Variance component	Proportional variance	$\hat{\sigma}^2$	$\hat{\sigma}^2$ p-v	alue	$h^2$ Maternal effect
Donor	0.000	0.000	0.00	1.000	0	
Recipient	1.988	0.061	8.98	0.003		0.061
Residual	30.840	0.939				

## SELECTION

### Data exploration

```
# OUTLIERS?
par(mfrow = c(1,2), mar = c(3, 4, 1, 1))
dotchart(S2$std.FFD, main= "standardised FFD") #OK
#dotchart(S2$volume, main= "volume")
dotchart(S2$rel.seeds, main= "Rel. Number of seeds") #nja
```



```
# ZERO INFLATION?
100 * sum(S2$rel.seeds == 0) / nrow(S2) #No

## [1] 22.43478

#inflation of low numbers of seeds?
#100 * length(unique(S2$PlantID[S2$N.seeds < 2])) / nrow(S2) #No
```

Data is not balanced:

```
#Balanced data?
table(S2$Dad) #
```

```
##
##      10:3:1      11:1:2      11:3:1      11:5:3      11:6:1
##          3          16          8          5          2
##      11:8:1      12:10:2      12:3:1      12:7:5      12:8:2
```

##	11	6	5	3	6
##	12:9:2	13:6:1	13:8:1	13:8:4	14:2:2
##	6	9	6	12	11
##	14:5:5	15:3:2	16:1:1	16:1:2	16:2:1
##	13	6	5	19	2
##	18:4:2	19:1:7	19:3:4	19:6:2	19:6:5
##	3	7	14	9	2
##	2:4:1	21:1:2	21:4:1	22:4:1	23:1:2
##	5	2	20	12	4
##	23:1:4	24:2:1	24:3:2	24:4:4	24:5:1
##	5	6	7	4	5
##	27:2:1	27:4:4	3:8:4	31:1:2	31:3:4
##	5	10	10	25	8
##	31:3:5	31:4:6	32:2:2	35:2:2	39:2:3
##	11	4	6	7	15
##	39:3:3	39:4:2	39:5:2	39:5:3	4:1:4
##	19	7	5	2	16
##	5:1:1	6:1:2	6:2:3	6:4:2	6:5:1
##	4	10	12	8	6
##	7:1:1	8:3:1	8:3:2	8:4:1	8:5:1
##	8	7	28	8	26
##	9:1:6	9:6:2	9:7:3	98_30_1:4:1	T2
##	8	20	13	8	

```
range(table(S2$Dad))
```

```
## [1] 2 28
```

```
table(S2$Mum)
```

##				
##	10:3:1	11:5:3	11:6:3	11:8:1
##	9	2	2	2
##	12:10:2	12:3:1	12:3:4	12:4:5
##	7	7	13	12
##	12:4:6	12:6:1	12:7:5	12:8:2
##	14	9	3	24
##	12:9:2	129_25_2:4:2	T2	13:6:2
##	16	10	10	13:7:1
##	13:8:4	14:2:2	15:3:1	9
##	2	13	21	15:3:2
##	16:1:1	16:5:1	19:1:4	5
##	9	10	22	19:1:7
##	19:2:4	19:4:2	19:5:1	3
##	2	3	7	19:5:2
##	19:5:5	19:6:5	2:1:3	7
##	19	9	27	2:4:1
##	21:2:4	21:3:3	21:5:3	12
##	7	2	8	22:4:1
##	24:1:1	32:2:2	32:5:1	15
##	2	3	32	32:8:3
##	33:1:1	36:3:3	38:2:1	21
##	6	8	14	39:2:2
##	39:3:2	39:3:3	39:4:2	5
##	8	11	15	39:5:2
##				37

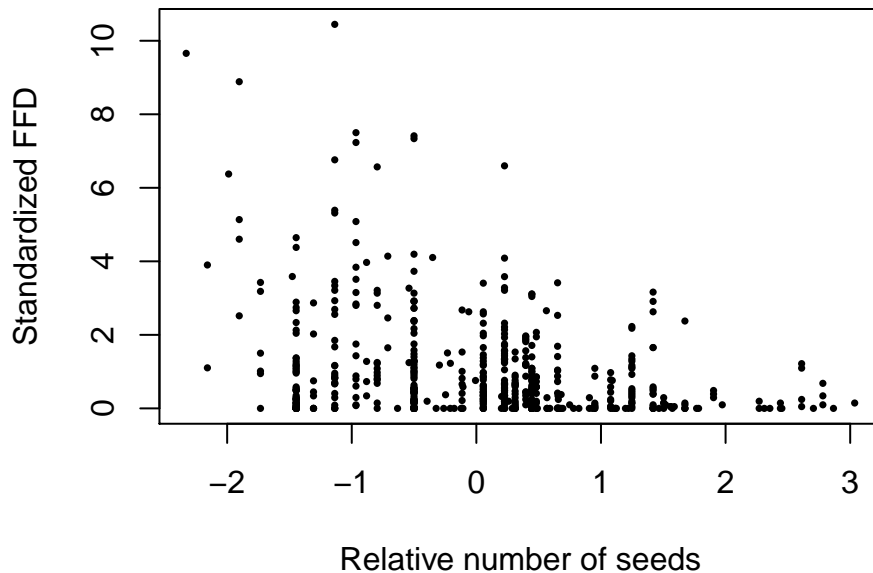
```
##          39:5:3          4:1:6          6:1:1          6:2:3
##             9             8             15             15
##          8:1:2          9:7:3
##             4             10
```

```
range(table(S2$Mum))
```

```
## [1]  2 37
```

```
# RELATIONSHIPS?
```

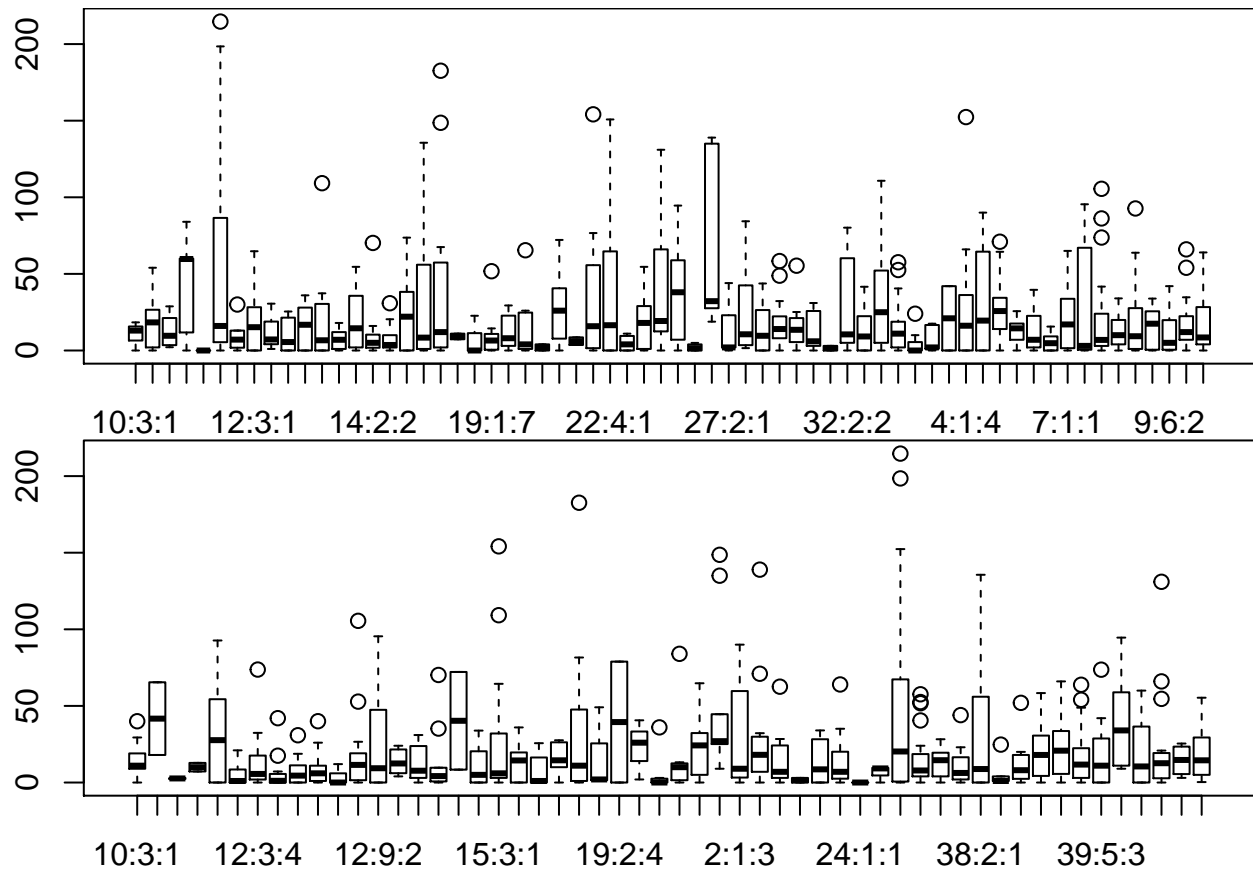
```
plot(x = S2$std.FFD,
     y = S2$rel.seeds,
     pch = 16, cex = 0.5,
     ylab = "Standardized FFD",
     xlab = "Relative number of seeds")
```



```
#par(mfrow=c(2,1), mar=c(2,2,0,0))
#boxplot(FFD ~ Dad, data = S2)
#boxplot(FFD ~ Mum, data = S2)
```

```
{par(mfrow=c(2,1), mar=c(2,2,0,0))
boxplot(N.seeds ~ Dad, data = S2)
boxplot(N.seeds ~ Mum, data = S2)}
```





## Construct model

```
library(MCMCglmm)
```

```
## Loading required package: coda
```

```
## Loading required package: ape
```

## Prior and model formulation

```
#library(parallel) #to run multiple chains

#####
# Parameter expanded prior
#####

# G = random effects
# 2 element (covar-matriser)(1 varians-vector hashed out)
# R = residual
#
# V = (co)variance limit?
# "Without loss of generality, V can be set to one"
#> diag(2)
# [ ,1] [ ,2]
```

```

# [1,] 1 0
# [2,] 0 1
#
# nu = degree of belief parameter (hur mkt jag tror p? prior?)
# alpha.mu = prior means: mean variance for parameter expanded prior
# alpha.v = prior covariance matrix

prior<-list(G=list(G1=list(V=diag(2), nu=3, alpha.mu=c(0,0), alpha.v=diag(2)*1000),
                  G2=list(V=diag(2), nu=3, alpha.mu=c(0,0), alpha.v=diag(2)*1000)),
            R=list(V=diag(2), nu=3))

#####
# Model formulation
#####

## FIXED EFFECT: ~ trait -1
##
## 1. Hadfield:
## Fit the fixed effect trait so that the two vars can
## have different intercepts.
## suppress the intercept (-1) so the second coefficient is not the
## difference between the intercepts of the two vars
## but the actual trait specific intercepts.
##
## 2. Wilson et al:
## Mean of trait instead of contrast between the two traits

mod1<-MCMCglmm(cbind(std.FFD, rel.seeds) ~ trait - 1,

random=~us(trait):Dad +
      us(trait):Mum ,
rcov=~us(trait):units, #RESID variance
family=c("gaussian", "gaussian"), #for each resp.var.
burnin=10000,
nitt=1010000,
thin=500,
data=S2,
prior=prior,
verbose = FALSE)

save(mod1, file="mod1.RData")

## changing from ~idh to ~us for resid var (rcov)
## --> lower (co)var estimates
## as that covar goes gen var if not accounted for

## Fitness often poisson distributed

```

```
## When mu is large the poisson distr --> normal

##Multiple chains: https://github.com/tmalsburg/MCMCglmm-intro
#set.seed(1)
#m6 <- mclapply(1:4, function(i) {
#  MCMCglmm(pronoun ~ (a + b + c)~3,
#    ~us(1 + a : b : c):subject +
#    us(1 + a : b) :item,
#    data = d,
#    family = "categorical",
#    prior = prior.m5,
#    thin = 20,
#    burnin = 3000,
#    nitt = 23000)
#}, mc.cores=4)

#m6 <- lapply(m6, function(m) m$Sol)
#m6 <- do.call(mcmc.list, m6)
```

## Re-run with other prior

```
#####
# Parameter expanded prior
#####

# G = random effects
# 2 element (covar-matriser)(1 varians-vector hashed out)
# R = residual
#
# V = (co)variance limit?
# "Without loss of generality, V can be set to one"
#> diag(2)
#      [,1] [,2]
#[1,]    1    0
#[2,]    0    1
#
# nu = degree of belief parameter (hur mkt jag tror p? prior?)
# alpha.mu = prior means: mean variance for parameter expanded prior
# alpha.v = prior covariance matrix

var_FFD <- var(S2$std.FFD)
var_seed <- var(S2$rel.seeds)

prior2<-list(G=list(G1=list(V=diag(2), nu=3, alpha.mu=c(var_FFD,var_seed), alpha.v=diag(2)*1000),
                    G2=list(V=diag(2), nu=3, alpha.mu=c(var_FFD,var_seed), alpha.v=diag(2)*1000)),
            R=list(V=diag(2), nu=3))

#####
# Model formulation
#####
```

```

## FIXED EFFECT: ~ trait -1
##
## 1. Hadfield:
## Fit the fixed effect trait so that the two vars can
## have different intercepts.
## suppress the intercept (-1) so the second coefficient is not the
## difference between the intercepts of the two vars
## but the actual trait specific intercepts.
##
## 2. Wilson etal:
## Mean of trait instead of contrast between the two traits

mod2<-MCMCglmm(cbind(std.FFD, rel.seeds) ~ trait - 1,

## RANDOM EFFECTS:
## us=unstructured, fits var/trait & covar traits
## us relaxes assumption of independence of groups
## us as both resp.vars measured for same inds

      random=~us(trait):Dad +
      us(trait):Mum ,
      rcov=~us(trait):units, #RESID variance
      family=c("gaussian", "gaussian"), #for each resp.var.
      burnin=10000,
      nitt=1010000,
      thin=500,
      data=S2,
      prior=prior2,
      verbose = FALSE)

save(mod2, file="mod2.RData")

## changing from ~idh to ~us for resid var (rcov)
## --> lower (co)var estimates
## as that covar goes gen var if not accounted for

## Fitness often poisson distributed
## When mu is large the poisson distr --> normal

```

## Inspect model

```

load("mod1.RData")

## AUTOCORRELATION
## OK if 2nd component between -0.1 and 0.1

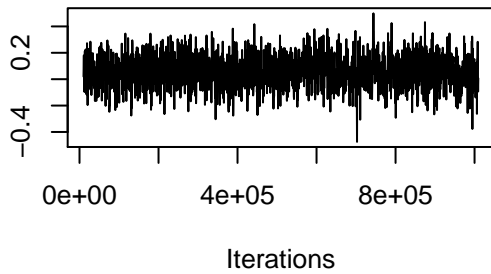
AC <- as.data.frame(autocorr(mod1$VCV))
range(AC["Lag 500",])

```

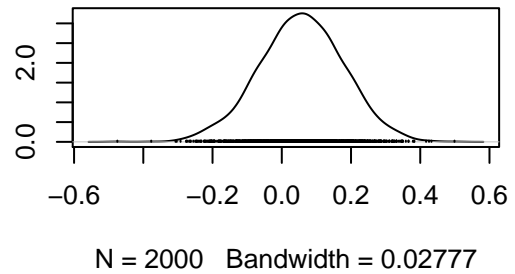
```
## [1] -0.06016630 0.05672929
```

```
##$Sol contains the distribution for the mean  
plot(mod1$Sol)
```

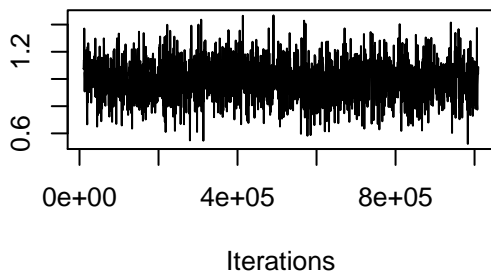
**Trace of traitstd.FFD**



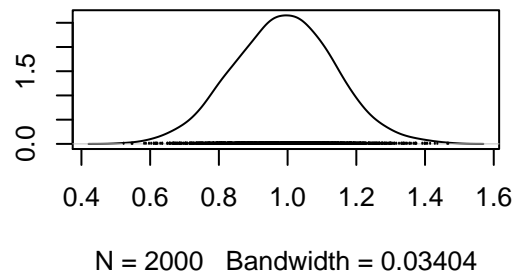
**Density of traitstd.FFD**



**Trace of traitrel.seeds**

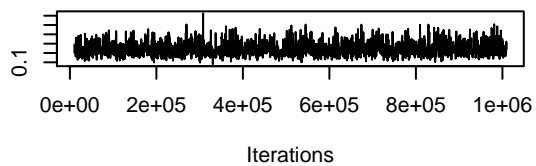


**Density of traitrel.seeds**

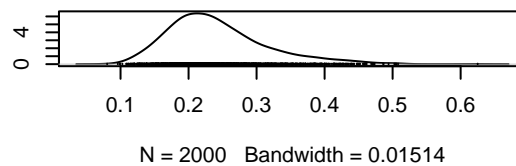


```
##$VCV contains the distribution for the variance  
plot(mod1$VCV)
```

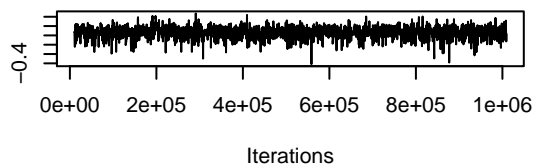
**Trace of traitstd.FFD:traitstd.FFD.Dad**



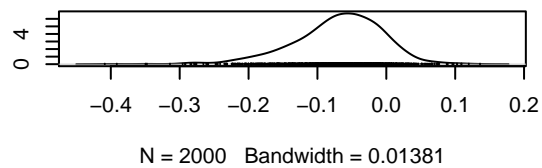
**Density of traitstd.FFD:traitstd.FFD.Dad**



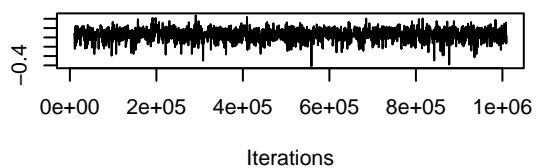
**Trace of traitrel.seeds:traitstd.FFD.Dad**



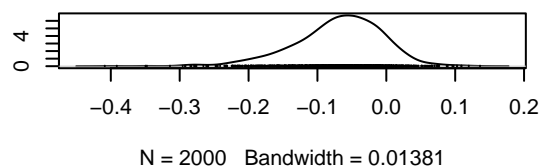
**Density of traitrel.seeds:traitstd.FFD.Dad**



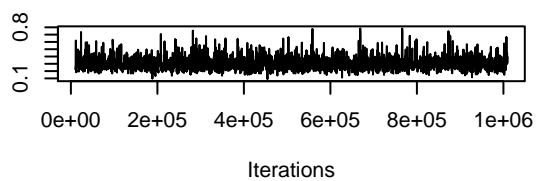
**Trace of traitstd.FFD:traitrel.seeds.Dad**



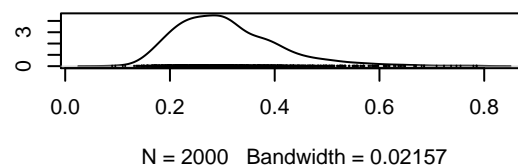
**Density of traitstd.FFD:traitrel.seeds.Dad**



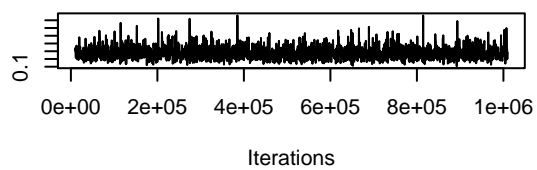
**Trace of traitrel.seeds:traitrel.seeds.Dad**



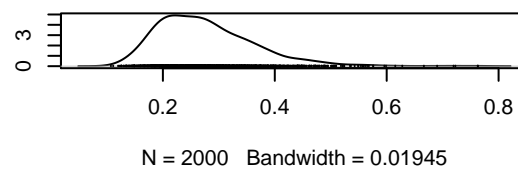
**Density of traitrel.seeds:traitrel.seeds.Dad**



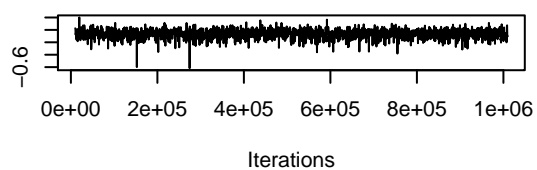
**Trace of traitstd.FFD:traitstd.FFD.Mum**



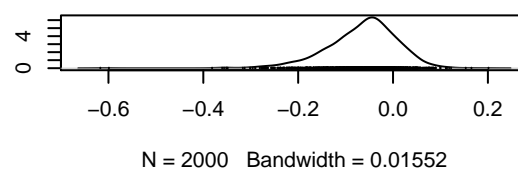
**Density of traitstd.FFD:traitstd.FFD.Mum**



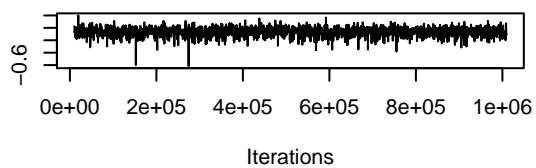
**Trace of traitrel.seeds:traitstd.FFD.Mum**



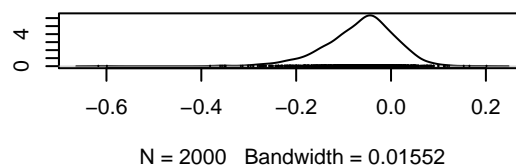
**Density of traitrel.seeds:traitstd.FFD.Mum**



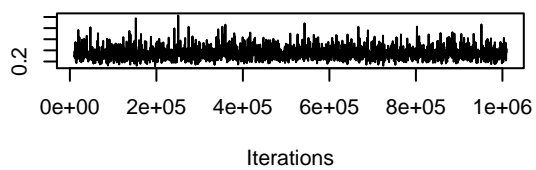
**Trace of traitstd.FFD:traitrel.seeds.Mum**



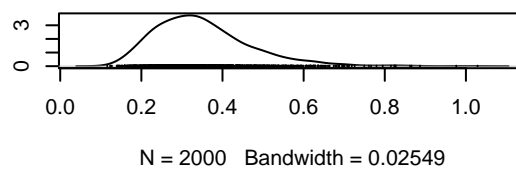
**Density of traitstd.FFD:traitrel.seeds.Mum**



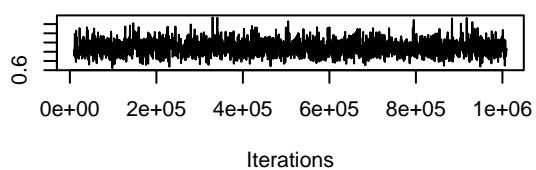
**Trace of traitrel.seeds:traitrel.seeds.Mum**



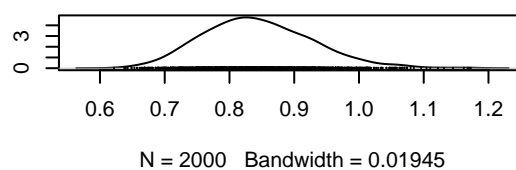
**Density of traitrel.seeds:traitrel.seeds.Mum**



**Trace of traitstd.FFD:traitstd.FFD.units**

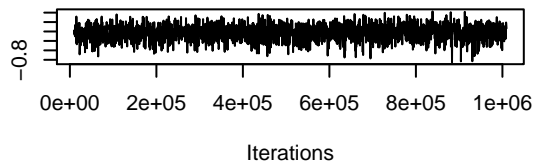


**Density of traitstd.FFD:traitstd.FFD.units**

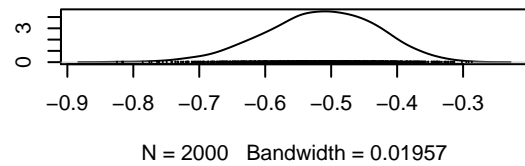




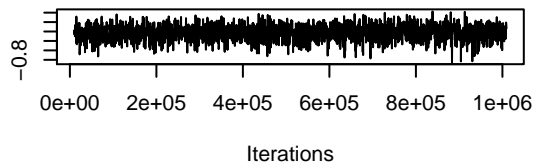
Trace of traitrel.seeds:traitstd.FFD.units



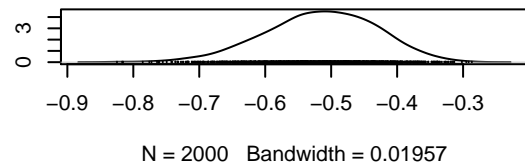
Density of traitrel.seeds:traitstd.FFD.units



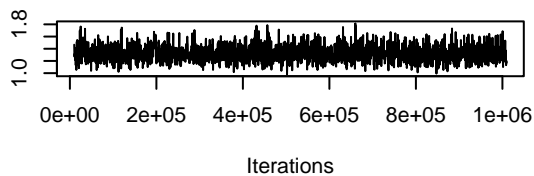
Trace of traitstd.FFD:traitrel.seeds.units



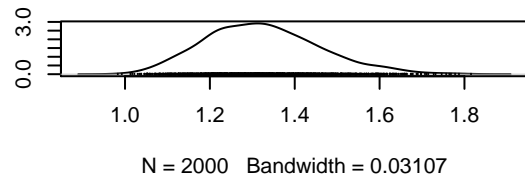
Density of traitstd.FFD:traitrel.seeds.units



Trace of traitrel.seeds:traitrel.seeds.units



Density of traitrel.seeds:traitrel.seeds.units



## Compare models

```
load("mod2.RData")

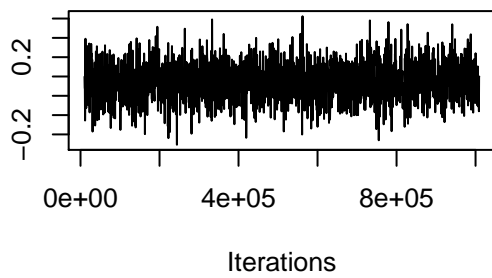
## AUTOCORRELATION
## OK if 2nd component between -0.1 and 0.1

AC <- as.data.frame(autocorr(mod2$VCV))
range(AC["Lag 500",])

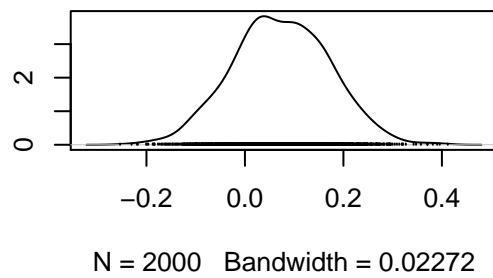
## [1] -0.05314411 0.05014882

##$Sol contains the distribution for the mean
plot(mod2$Sol)
```

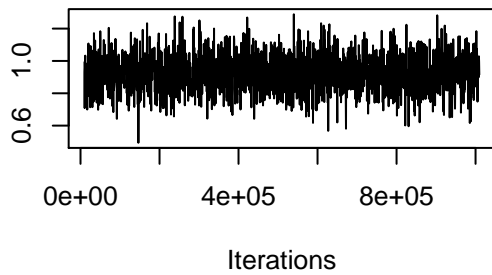
**Trace of traitstd.FFD**



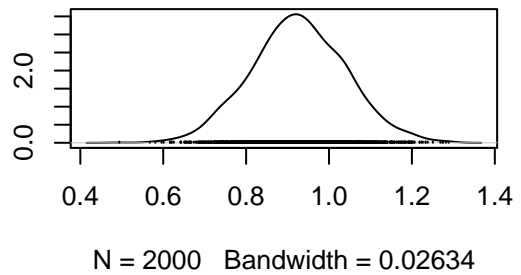
**Density of traitstd.FFD**



**Trace of traitrel.seeds**

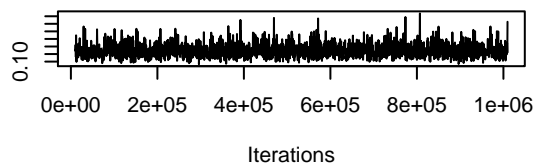


**Density of traitrel.seeds**

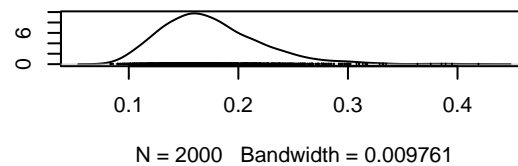


```
#$VCV contains the distribution for the variance  
plot(mod2$VCV)
```

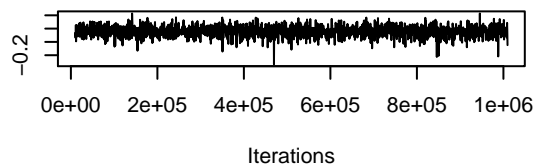
**Trace of traitstd.FFD:traitstd.FFD.Dad**



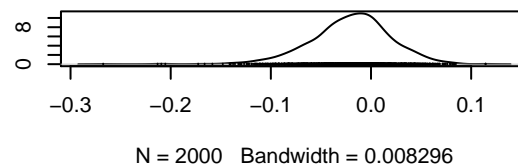
**Density of traitstd.FFD:traitstd.FFD.Dad**



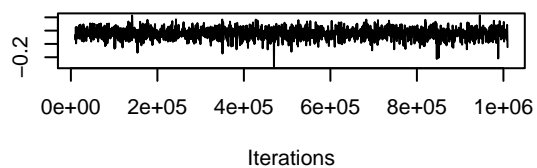
**Trace of traitrel.seeds:traitstd.FFD.Dad**



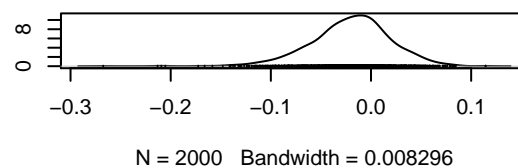
**Density of traitrel.seeds:traitstd.FFD.Dad**



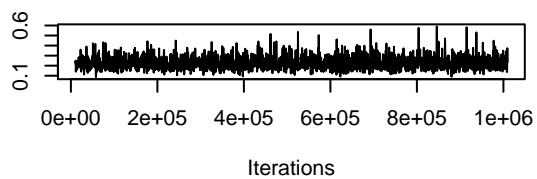
**Trace of traitstd.FFD:traitrel.seeds.Dad**



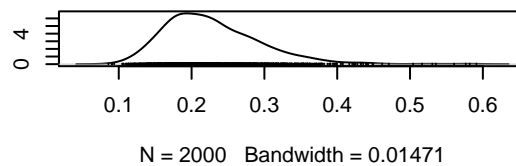
**Density of traitstd.FFD:traitrel.seeds.Dad**



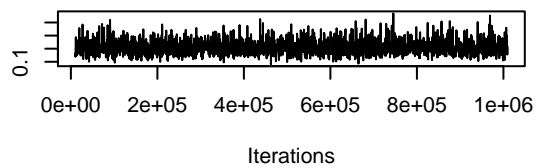
**Trace of traitrel.seeds:traitrel.seeds.Dad**



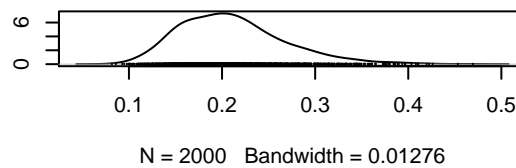
**Density of traitrel.seeds:traitrel.seeds.Dad**



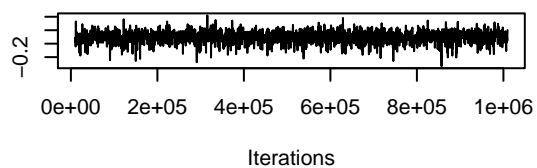
**Trace of traitstd.FFD:traitstd.FFD.Mum**



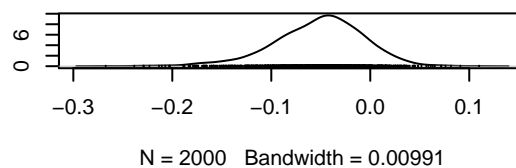
**Density of traitstd.FFD:traitstd.FFD.Mum**



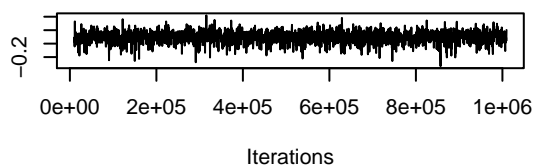
**Trace of traitrel.seeds:traitstd.FFD.Mum**



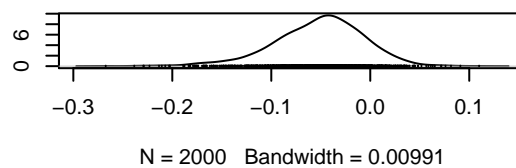
**Density of traitrel.seeds:traitstd.FFD.Mum**



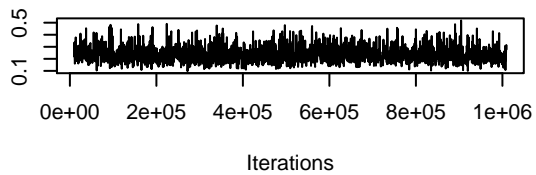
**Trace of traitstd.FFD:traitrel.seeds.Mum**



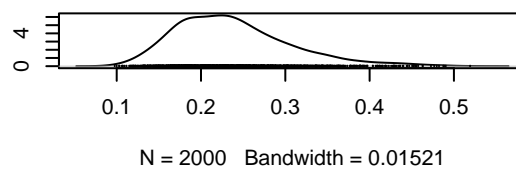
**Density of traitstd.FFD:traitrel.seeds.Mum**



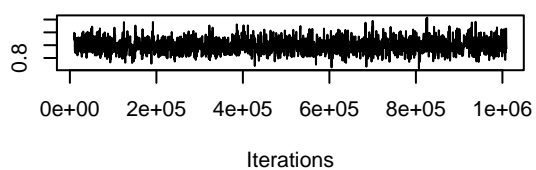
**Trace of traitrel.seeds:traitrel.seeds.Mum**



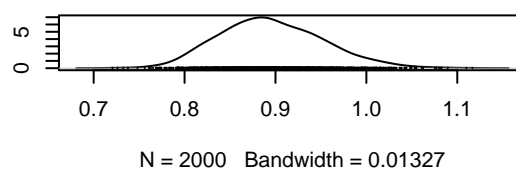
**Density of traitrel.seeds:traitrel.seeds.Mum**

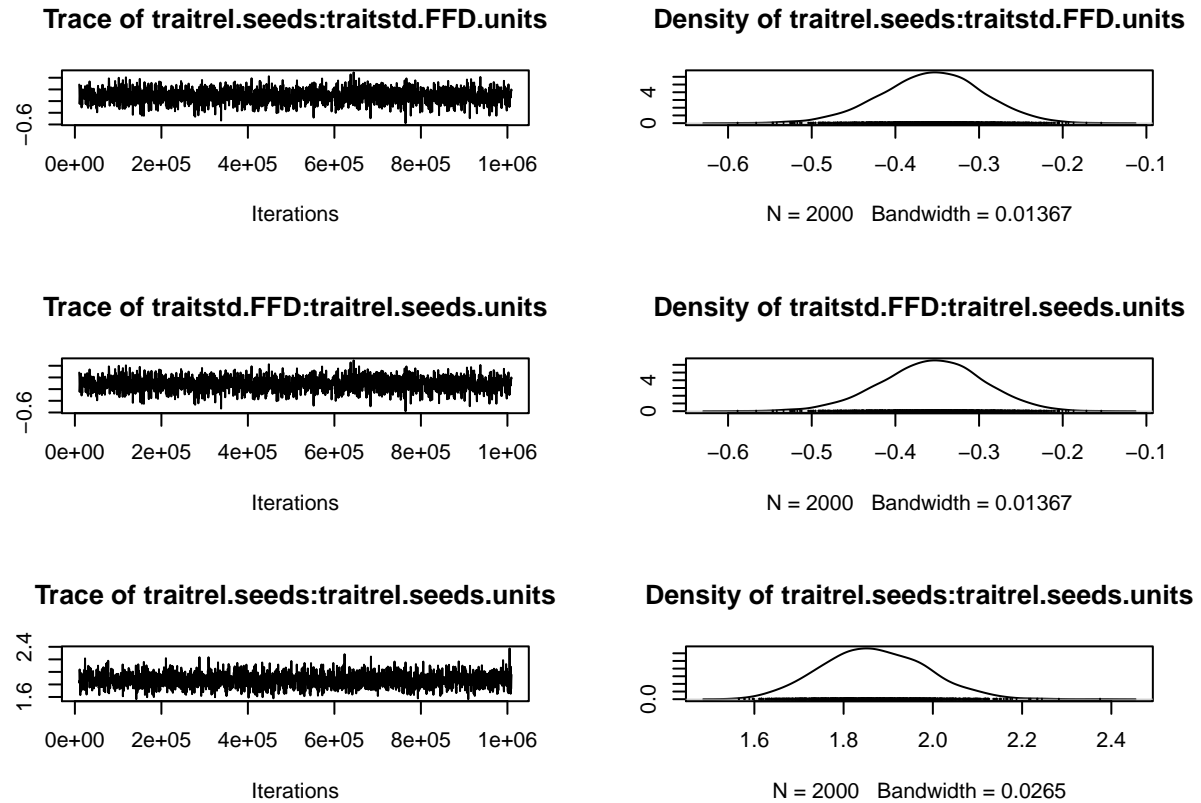


**Trace of traitstd.FFD:traitstd.FFD.units**



**Density of traitstd.FFD:traitstd.FFD.units**





## Model summary

```
summary(mod1) # should be large sample size for all vars & covars
```

```
##
## Iterations = 10001:1009501
## Thinning interval = 500
## Sample size = 2000
##
## DIC: 1414.568
##
## G-structure: ~us(trait):Dad
##
##               post.mean l-95% CI u-95% CI eff.samp
## traitstd.FFD:traitstd.FFD.Dad      0.24158   0.1198  0.39554     2000
## traitrel.seeds:traitstd.FFD.Dad    -0.06942  -0.2064  0.04196     2000
## traitstd.FFD:traitrel.seeds.Dad    -0.06942  -0.2064  0.04196     2000
## traitrel.seeds:traitrel.seeds.Dad   0.30934   0.1450  0.50628     2219
##
##               ~us(trait):Mum
##
##               post.mean l-95% CI u-95% CI eff.samp
## traitstd.FFD:traitstd.FFD.Mum      0.27664   0.1426  0.45367     2000
## traitrel.seeds:traitstd.FFD.Mum    -0.06463  -0.2251  0.05869     2132
## traitstd.FFD:traitrel.seeds.Mum    -0.06463  -0.2251  0.05869     2132
```

```

## traitrel.seeds:traitrel.seeds.Mum    0.35034    0.1579    0.59211    2000
##
## R-structure: ~us(trait):units
##
##
## post.mean l-95% CI u-95% CI eff.samp
## traitstd.FFD:traitstd.FFD.units      0.8441    0.6966    1.0166    2000
## traitrel.seeds:traitstd.FFD.units     -0.5156   -0.6800   -0.3572    2000
## traitstd.FFD:traitrel.seeds.units     -0.5156   -0.6800   -0.3572    2000
## traitrel.seeds:traitrel.seeds.units    1.3239    1.0808    1.5980    1818
##
## Location effects: cbind(std.FFD, rel.seeds) ~ trait - 1
##
## post.mean l-95% CI u-95% CI eff.samp pMCMC
## traitstd.FFD      0.05652 -0.18205  0.29896    2000  0.621
## traitrel.seeds    0.98924  0.70242  1.28266    2000 <5e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#
#---
# Iterations = 10001:1009501
# Thinning interval = 500
# Sample size = 2000
#
# DIC: 1414.568
#
# G-structure: ~us(trait):Dad #RANDOM EFFECT Dad (additive gen effects?)
#
# Variances can't be negative (hard to know if important)
# Covariances overlap 0 = not important
#
# post.mean l-95% CI u-95% CI eff.samp
#traitstd.FFD:traitstd.FFD.Dad      0.24158    0.1198    0.39554    2000
#traitrel.seeds:traitstd.FFD.Dad     -0.06942   -0.2064    0.04196    2000
#traitstd.FFD:traitrel.seeds.Dad     -0.06942   -0.2064    0.04196    2000
#traitrel.seeds:traitrel.seeds.Dad    0.30934    0.1450    0.50628    2219
#
# ~us(trait):Mum #RANDOM EFFECT Mum
# Covariances overlap 0
#
# post.mean l-95% CI u-95% CI eff.samp
#traitstd.FFD:traitstd.FFD.Mum      0.27664    0.1426    0.45367    2000
#traitrel.seeds:traitstd.FFD.Mum     -0.06463   -0.2251    0.05869    2132
#traitstd.FFD:traitrel.seeds.Mum     -0.06463   -0.2251    0.05869    2132
#traitrel.seeds:traitrel.seeds.Mum    0.35034    0.1579    0.59211    2000
#
# R-structure: ~us(trait):units #RESIDUALS
# covariance not overlapping 0 => they are important
# Selection is on individuals, not genetic
#
# post.mean l-95% CI u-95% CI eff.samp
#traitstd.FFD:traitstd.FFD.units      0.8441    0.6966    1.0166    2000
#traitrel.seeds:traitstd.FFD.units     -0.5156   -0.6800   -0.3572    2000

```

```

#traitstd.FFD:traitrel.seeds.units      -0.5156  -0.6800  -0.3572    2000
#traitrel.seeds:traitrel.seeds.units      1.3239   1.0808   1.5980    1818

# Location effects: cbind(std.FFD, rel.seeds) ~ trait - 1
#
#               post.mean l-95% CI u-95% CI eff.samp  pMCMC
#traitstd.FFD      0.05652 -0.18205  0.29896     2000  0.621
#traitrel.seeds    0.98924  0.70242  1.28266     2000 <5e-04 ***
#---
#Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
summary(mod2)
```

```

##
## Iterations = 10001:1009501
## Thinning interval = 500
## Sample size = 2000
##
## DIC: 3632.601
##
## G-structure: ~us(trait):Dad
##
##               post.mean l-95% CI u-95% CI eff.samp
## traitstd.FFD:traitstd.FFD.Dad      0.17464  0.09606  0.26095    1562
## traitrel.seeds:traitstd.FFD.Dad    -0.02002 -0.09843  0.05847    2000
## traitstd.FFD:traitrel.seeds.Dad    -0.02002 -0.09843  0.05847    2000
## traitrel.seeds:traitrel.seeds.Dad   0.22828  0.12040  0.35933    2343
##
##               ~us(trait):Mum
##
##               post.mean l-95% CI u-95% CI eff.samp
## traitstd.FFD:traitstd.FFD.Mum      0.21088  0.1081  0.3209    2000
## traitrel.seeds:traitstd.FFD.Mum    -0.05175 -0.1438  0.0348    2000
## traitstd.FFD:traitrel.seeds.Mum    -0.05175 -0.1438  0.0348    2000
## traitrel.seeds:traitrel.seeds.Mum   0.23777  0.1148  0.3704    2000
##
## R-structure: ~us(trait):units
##
##               post.mean l-95% CI u-95% CI eff.samp
## traitstd.FFD:traitstd.FFD.units      0.8957  0.7915  1.0084    2000
## traitrel.seeds:traitstd.FFD.units    -0.3564 -0.4745 -0.2395    2000
## traitstd.FFD:traitrel.seeds.units    -0.3564 -0.4745 -0.2395    2000
## traitrel.seeds:traitrel.seeds.units   1.8693  1.6444  2.0858    2000
##
## Location effects: cbind(std.FFD, rel.seeds) ~ trait - 1
##
##               post.mean l-95% CI u-95% CI eff.samp  pMCMC
## traitstd.FFD      0.07315 -0.12430  0.24997     2024  0.455
## traitrel.seeds    0.92614  0.69254  1.13741     1992 <5e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```



```

#koll
#beta = covar(trait, fitness) / var(trait)

#var(S2$rel.seeds)
#var(S2$std.FFD)
#cor(S2$rel.seeds,S2$std.FFD)

#cor(S2$rel.seeds,S2$std.FFD)* (var(S2$rel.seeds)/var(S2$std.FFD))

#beta = var(cov(a/b)*(var_a/var_b))

```

## Selection gradients

```

#colnames(mod1$VCV)
#head(mod1$VCV)

#Beta.ad
#Beta.dam
#Beta.resid

#Beta=covar(FFD, Seeds)/Var(FFD)

# Additive gen effects
cov.a<-mod1$VCV[, "traitstd.FFD:traitrel.seeds.Dad"]
var.a.FFD<-mod1$VCV[, "traitstd.FFD:traitstd.FFD.Dad"]
BetaDad <- cov.a/var.a.FFD
mean(BetaDad)

## [1] -0.2762674

HPDinterval(BetaDad)

##          lower      upper
## var1 -0.7075401 0.192543
## attr(,"Probability")
## [1] 0.95

# Maternal effects: Should I subtract donor effects!?!?
cov.m<-mod1$VCV[, "traitstd.FFD:traitrel.seeds.Mum"]
var.m.FFD<-mod1$VCV[, "traitstd.FFD:traitstd.FFD.Mum"]
BetaMum <- cov.m/var.m.FFD
mean(BetaMum)

## [1] -0.224464

HPDinterval(BetaMum)

##          lower      upper
## var1 -0.6851692 0.2549093
## attr(,"Probability")
## [1] 0.95

# Residual, individual effects
cov.i<-mod1$VCV[, "traitstd.FFD:traitrel.seeds.units"]
var.i.FFD<-mod1$VCV[, "traitstd.FFD:traitstd.FFD.units"]

```

```

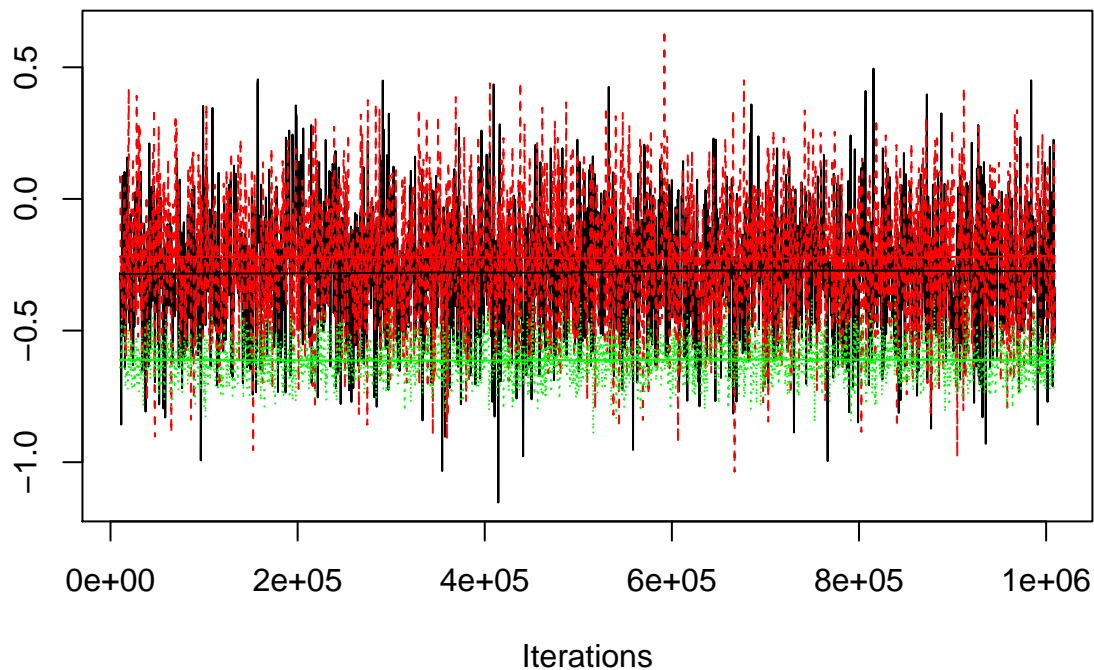
BetaInd <- cov.i/var.i.FFD
mean(BetaInd)

## [1] -0.6110475
HPDinterval(BetaInd)

##           lower      upper
## var1 -0.7528657 -0.4480618
## attr(,"Probability")
## [1] 0.95

plot(mcmc.list(BetaDad, BetaMum, BetaInd),col=c("black", "red", "green"), density = FALSE)

```



## Table

```

Gcov <- as.data.frame(summary(mod1)$Gcovariances)
Gcov <- Gcov[c(2,6),]
Rcov <- as.data.frame(summary(mod1)$Rcovariances)
Rcov <- Rcov[2,]

COV <- rbind(Gcov, Rcov)
rownames(COV) <- c("Donor", "Recipient", "Residual")

Beta <- c(mean(BetaDad), mean(BetaMum), mean(BetaInd))
HPDInt <- matrix(data=c(HPDinterval(BetaDad), HPDinterval(BetaMum), HPDinterval(BetaInd)),nrow=3, ncol=2)

```

```
Betas <- as.data.frame(cbind(Beta, HPDInt))
rownames(Betas) <- c("Donor", "Recipient", "Residual")
colnames(Betas) <- c("Beta", "HPD interval")
knitr::kable(Betas)
```

	Beta	HPD interval	NA
Donor	-0.2762674	-0.7075401	0.1925430
Recipient	-0.2244640	-0.6851692	0.2549093
Residual	-0.6110475	-0.7528657	-0.4480618

```
mytable <- cbind(COV, Betas)
```

```
knitr::kable(mytable, digits = 3)
```

	post.mean	l-95% CI	u-95% CI	eff.samp	Beta	HPD interval	NA
Donor	-0.069	-0.206	0.042	2000.000	-0.276	-0.708	0.193
Recipient	-0.065	-0.225	0.059	2132.226	-0.224	-0.685	0.255
Residual	-0.516	-0.680	-0.357	2000.000	-0.611	-0.753	-0.448

### Skillnad i gradienter?

```
BetaDiff <- BetaInd - BetaDad
summary(BetaInd)
```

```
##
## Iterations = 10001:1009501
## Thinning interval = 500
## Number of chains = 1
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean           SD       Naive SE Time-series SE
##      -0.611047      0.080167      0.001793      0.001793
##
## 2. Quantiles for each variable:
##
##      2.5%      25%      50%      75%      97.5%
## -0.7608 -0.6645 -0.6129 -0.5570 -0.4541
```

```
summary(BetaDiff)
```

```
##
## Iterations = 10001:1009501
## Thinning interval = 500
## Number of chains = 1
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
```

```
##
##           Mean           SD      Naive SE Time-series SE
##      -0.334780      0.247500      0.005534      0.005534
##
## 2. Quantiles for each variable:
##
##      2.5%      25%      50%      75%      97.5%
## -0.8378 -0.5001 -0.3322 -0.1603  0.1325
summary(BetaDad)
```

```
##
## Iterations = 10001:1009501
## Thinning interval = 500
## Number of chains = 1
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean           SD      Naive SE Time-series SE
##      -0.276267      0.233240      0.005215      0.005215
##
## 2. Quantiles for each variable:
##
##      2.5%      25%      50%      75%      97.5%
## -0.7200 -0.4348 -0.2784 -0.1186  0.1821
```

```
HPDinterval(BetaInd)
```

```
##           lower      upper
## var1 -0.7528657 -0.4480618
## attr(,"Probability")
## [1] 0.95
```

```
HPDinterval(BetaDiff)
```

```
##           lower      upper
## var1 -0.8000217 0.1538523
## attr(,"Probability")
## [1] 0.95
```

```
HPDinterval(BetaDad)
```

```
##           lower      upper
## var1 -0.7075401 0.192543
## attr(,"Probability")
## [1] 0.95
```