

# Lab 1 - DIY fitting MLM

Alencar Xavier, Gota Morota

October 26, 2018

# Example 1 - Sorghum

# Example 1 - load data

Example dataset from Kevin's *agridat* package

```
data(adugna.sorghum, package = 'agridat')  
dt = adugna.sorghum  
head(dt)
```

```
##   gen trial env yield year   loc  
## 1 G16    T2 E01   590 2001 Mieso  
## 2 G17    T2 E01   554 2001 Mieso  
## 3 G18    T2 E01   586 2001 Mieso  
## 4 G19    T2 E01   738 2001 Mieso  
## 5 G20    T2 E01   489 2001 Mieso  
## 6 G21    T2 E01   684 2001 Mieso
```

# Example 1 - Getting design matrix

- Linear model:  $Pheotype = Env + Gen$
- In algebra notation:  $y = Xb + Zu + e$

```
y = dt$yield  
X = model.matrix(y~env,dt)  
Z = model.matrix(y~gen-1,dt) # "-1" means no intercept
```

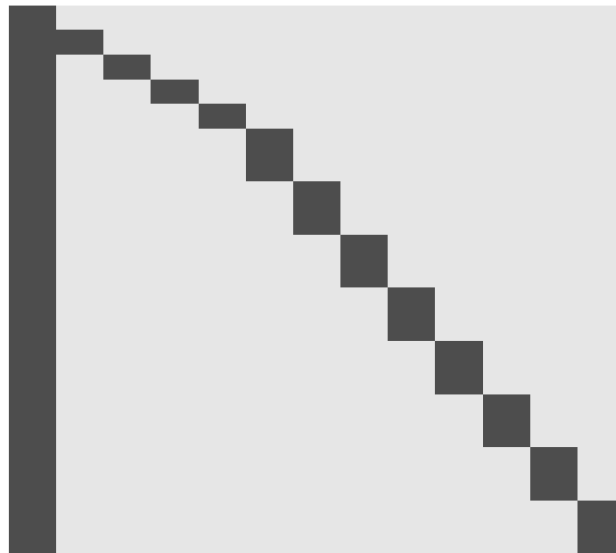
Assuming:

- $u \sim N(0, I\sigma_g^2)$
- $e \sim N(0, I\sigma_e^2)$

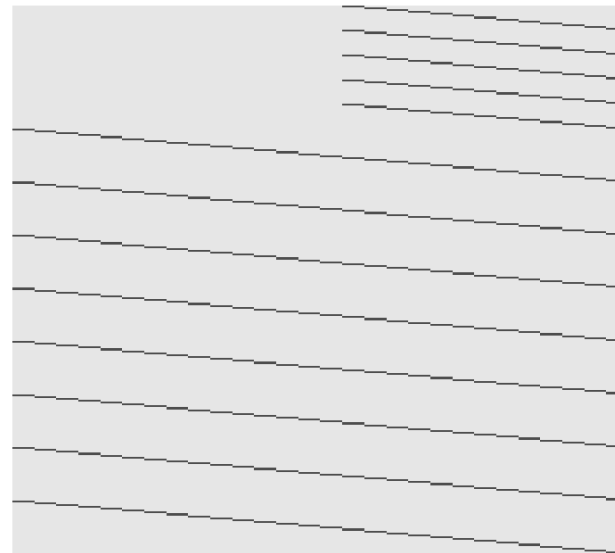
# Example 1 - Visualize X and Z matrices

```
SEE=function(A,...)image(t(1-A[nrow(A):1,]),axes=F,col=gray.colors(2),...)
par(mfrow=c(1,2),mar=c(1,1,3,1))
SEE(X, main=paste("X matrix (",paste(dim(X),collapse=' x '),")" ))
SEE(Z, main=paste("Z matrix (",paste(dim(Z),collapse=' x '),")" ))
```

**X matrix ( 289 x 13 )**



**Z matrix ( 289 x 28 )**



# Example 1 - Fit the model

```
# Using the NAM package (same for rrBLUP, EMMREML, BGLR)  
require(NAM, quietly = TRUE)  
fit1 = reml(y=y,X=X,Z=Z)
```

```
# Alternatively, you can also use formulas with NAM  
fit1b = reml(y=dt$yield,X=~dt$env,Z=~dt$gen )
```

```
# Using the lme4 package  
require(lme4, quietly = TRUE)  
fit2 = lmer(yield ~ env + (1|gen), data=dt)
```

# Example 1 - Variance components

```
fit1$VC[c(1:2)] # same with fit1b$VC
```

```
##           Vg           Ve
## 1 189680.4 442075.6
```

```
data.frame((summary(fit2))$varcor)$vcov
```

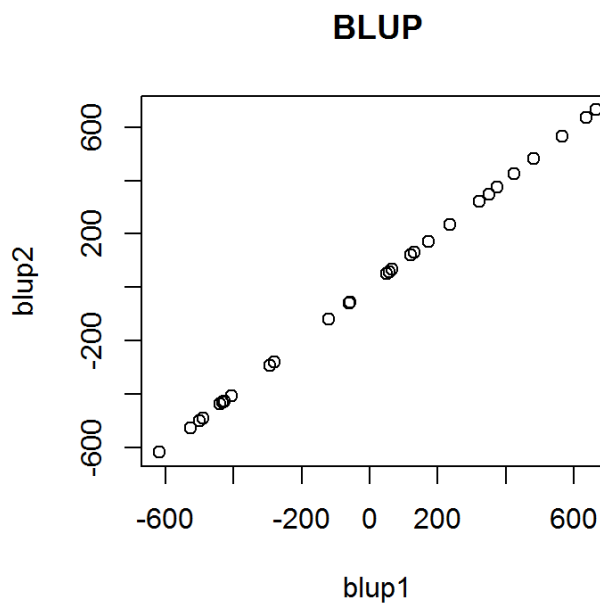
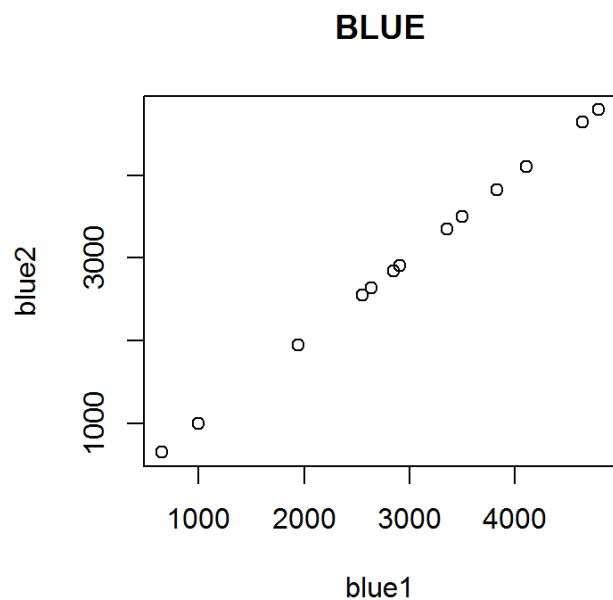
```
## [1] 189680.4 442075.6
```

- VC can be used to measure *broad-sense* heritability

$$H = \frac{\sigma_g^2}{\sigma_g^2 + \sigma_e^2/n} = \frac{189680.4}{189680.4 + 442075.6/10.32} = 0.82$$

# Example 1 - The coefficients

```
blue1 = fit1$Fixed[,1]; blup1 = fit1$EBV  
blue2 = fit2@beta;      blup2 = rowMeans(ranef(fit2)$gen)  
par(mfrow=c(1,2));  
plot(blue1,blue2,main="BLUE"); plot(blup1,blup2,main="BLUP")
```



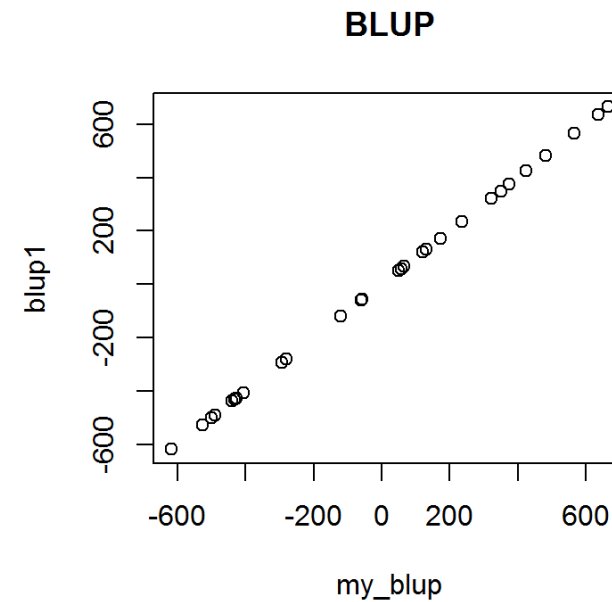
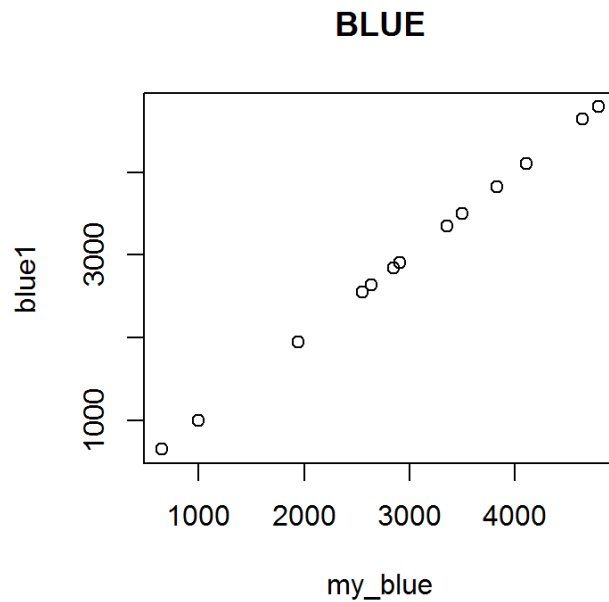


# Example 1 - DIY BLUPs

```
iK = diag(ncol(Z))
Lambda = 442075.6/189680.4
W = cbind(X,Z)
Sigma = as.matrix(Matrix::bdiag(diag(0,ncol(X)),iK*Lambda))
LHS = crossprod(W) + Sigma
RHS = crossprod(W,y)
g = solve(LHS,RHS)
my_blue = g[ c(1:ncol(X))]
my_blup = g[-c(1:ncol(X))]
```

# Example 1 - DIY BLUPs

```
par(mfrow=c(1,2))  
plot(my_blue,blue1,main="BLUE")  
plot(my_blup,blup1,main="BLUP")
```



# Example 1 - DIY Variance components

$$\sigma_e^2 = \frac{e'y}{n-p} \text{ and } \sigma_u^2 = \frac{u'K^{-1}u + \text{tr}(K^{-1}C^{22}\sigma_e^2)}{q}$$

```
e = y - X %% my_blue - Z %% my_blup
```

```
Ve = c(y%%e)/(length(y)-ncol(X))
```

```
Ve
```

```
## [1] 442075.6
```

```
trKC22 = sum(diag(iK%%(solve(LHS)[-c(1:ncol(X)), -c(1:ncol(X))])))
```

```
Vg = Vg = c(t(my_blup)%%iK%%my_blup+trKC22*Ve)/ncol(Z)
```

```
Vg
```

```
## [1] 189680.4
```

# Starting from bad variance components

```

Ve = Vg = 1
for(i in 1:25){
  Lambda = Ve/Vg;
  Sigma = as.matrix(Matrix::bdiag(diag(0,ncol(X)),iK*Lambda))
  LHS = crossprod(W) + Sigma; RHS = crossprod(W,y); g = solve(LHS,RHS)
  my_blue = g[ c(1:ncol(X))]; my_blup = g[-c(1:ncol(X))]
  e = y - X%*%my_blue - Z%*%my_blup; Ve = c(y%*%e)/(length(y)-ncol(X))
  trKC22 = sum(diag(iK%*(solve(LHS)[(ncol(X)+1):(ncol(W))),(ncol(X)+1):(ncol(W))]))
  Vg = c(t(my_blup)%*%iK%*my_blup+trKC22*Ve)/ncol(Z)
  if(!i%5){cat('It',i,'VC:  Vg = ',Vg,'and Ve = ',Ve,'\n')}}

## It 5 VC:  Vg = 191751.1 and Ve = 441110.7
## It 10 VC:  Vg = 189728.4 and Ve = 442053.2
## It 15 VC:  Vg = 189681.5 and Ve = 442075.1
## It 20 VC:  Vg = 189680.4 and Ve = 442075.6
## It 25 VC:  Vg = 189680.4 and Ve = 442075.6

```

# Example 2 - Barley

## Example 2 - load data

Another example dataset from Kevin's *agridat* package

```
data(stepToe.morex.pheno, package='agridat')  
dt = stepToe.morex.pheno  
head(dt)
```

##	gen	env	amylase	diapow	hddate	lodging	malt	height	protein	yield
## 1	StepToe	MN92	22.7	46	149.5	NA	73.6	84.5	10.5	5.5315
## 2	StepToe	MTi92	30.1	72	178.0	10	76.5	NA	11.2	8.6403
## 3	StepToe	MTd92	26.7	78	165.0	15	74.5	75.5	13.4	5.8990
## 4	StepToe	ID91	26.2	74	179.0	NA	74.1	111.0	12.1	8.6290
## 5	StepToe	OR91	19.6	62	191.0	NA	71.5	90.0	11.7	5.3440
## 6	StepToe	WA91	23.6	54	181.0	NA	73.8	112.0	10.0	6.2700

## Example 2 - Getting design matrix

- Linear model:  $Phe = Env + Gen$
- In algebra notation:  $y = Xb + Zu + e$

```
X = model.matrix(~env,dt)
Z = model.matrix(~gen-1,dt) # "-1" means no intercept
y = dt$yield
```

## Example 2 - Fit the model

```
# Fit
fit0 = reml(y=y,X=X,Z=Z)

# BLUE and BLUP
blue0 = fit0$Fixed[,1]
blup0 = fit0$EBV

# Get VC
fit0$VC[c(1:2)]

##           Vg           Ve
## 1 0.1320092 0.6379967
```

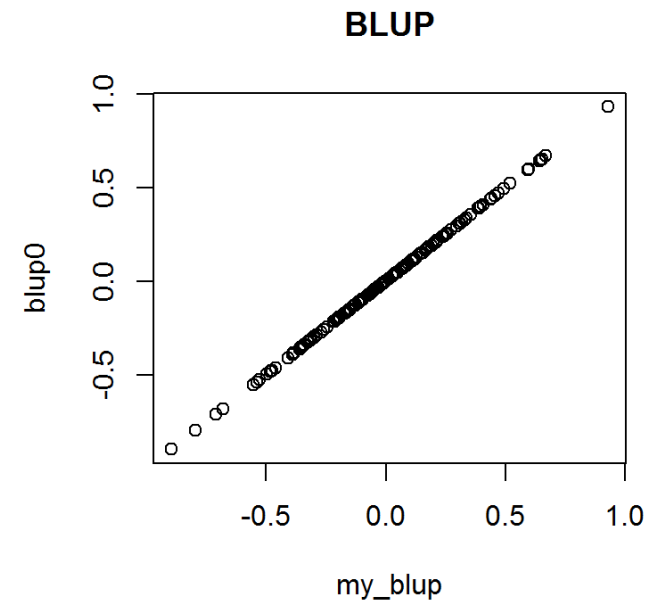
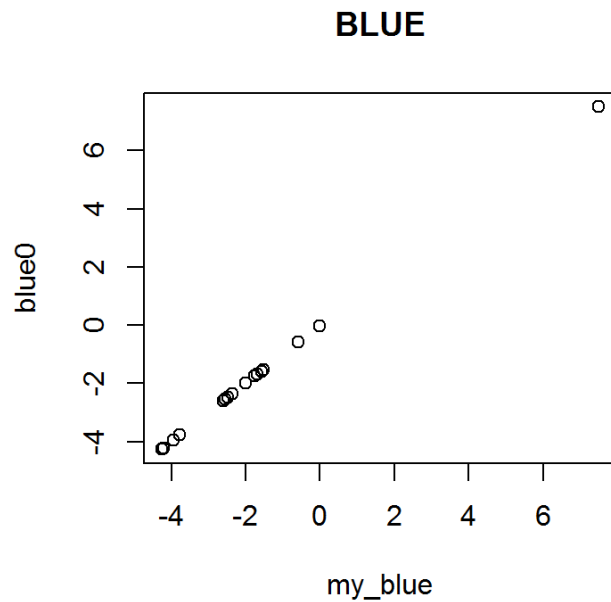


## Example 2 - DIY BLUPs

```
iK = diag(ncol(Z))
Lambda = 0.637997/0.132009
W = cbind(X,Z)
Sigma = as.matrix(Matrix::bdiag(diag(0,ncol(X)),iK*Lambda))
LHS = crossprod(W) + Sigma
RHS = crossprod(W,y)
g = solve(LHS,RHS)
my_blue = g[ c(1:ncol(X))]
my_blup = g[-c(1:ncol(X))]
```

## Example 2 - DIY BLUPs

```
par(mfrow=c(1,2))  
plot(my_blue,blue0,main="BLUE")  
plot(my_blup,blup0,main="BLUP")
```



## Example 2 - Check variance components

$$\sigma_e^2 = \frac{e'y}{n-p} \text{ and } \sigma_u^2 = \frac{u'K^{-1}u + \text{tr}(K^{-1}C^{22}\sigma_e^2)}{q}$$

```
e = y - X %% my_blue - Z %% my_blup
```

```
Ve = c(y%%e)/(length(y)-ncol(X))
```

```
Ve
```

```
## [1] 0.6379967
```

```
trKC22 = sum(diag(iK%%(solve(LHS)[-c(1:ncol(X)), -c(1:ncol(X))])))
```

```
Vg = c(t(my_blup)%%iK%%my_blup+trKC22*Ve)/ncol(Z)
```

```
Vg
```

```
## [1] 0.1320091
```

# Starting from bad variance components

```

Ve = Vg = 1
for(i in 1:25){
  Lambda = Ve/Vg;
  Sigma = as.matrix(Matrix::bdiag(diag(0,ncol(X)),iK*Lambda))
  LHS = crossprod(W) + Sigma; RHS = crossprod(W,y); g = solve(LHS,RHS)
  my_blue = g[ c(1:ncol(X))]; my_blup = g[-c(1:ncol(X))]
  e = y - X%*%my_blue - Z%*%my_blup; Ve = c(y%*%e)/(length(y)-ncol(X))
  trKC22 = sum(diag(iK%*(solve(LHS)[(ncol(X)+1):(ncol(W)),(ncol(X)+1):(ncol(W))]))))
  Vg = c(t(my_blup)%*%iK%*my_blup+trKC22*Ve)/ncol(Z)
  if(!i%5){cat('It',i,'VC:  Vg =',Vg,'and Ve =',Ve,'\n')}}

```

```

## It 5 VC:  Vg = 0.1336139 and Ve = 0.6370751
## It 10 VC:  Vg = 0.1320386 and Ve = 0.6379797
## It 15 VC:  Vg = 0.1320097 and Ve = 0.6379964
## It 20 VC:  Vg = 0.1320092 and Ve = 0.6379967
## It 25 VC:  Vg = 0.1320092 and Ve = 0.6379967

```

# Example 3 - Barley GEBV

# Using genomic information!

```
data(stepToe.morex.geno, package='agridat')
gen = do.call("cbind", lapply(stepToe.morex.geno$geno, function(x) x$data))
gen = rbind(0, 2, gen)
rownames(gen) = c('Morex', 'StepToe', as.character(stepToe.morex.geno$pheno$gen))
rownames(gen)[10] = "SM8"
gen = gen[gsub('gen', '', colnames(Z)), ]
K = G2A_Kernels(gen)$A
```

## Example 3 - Fit the model

```
# Fit model
```

```
fit0 = reml(y=y,X=X,Z=Z,K=K)
```

```
# BLUE and BLUP
```

```
blue0 = fit0$Fixed[,1]
```

```
gebv0 = fit0$EBV
```

```
# Get VC
```

```
fit0$VC[c(1:2)]
```

```
##           Vg           Ve
```

```
## 1 0.2334026 0.6575786
```

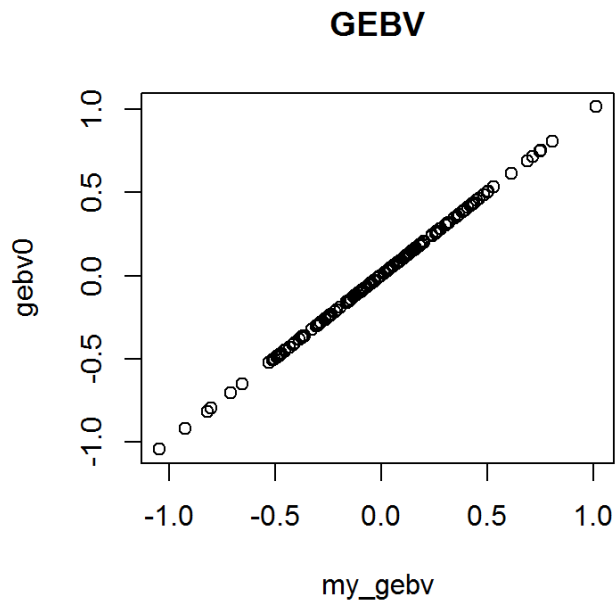
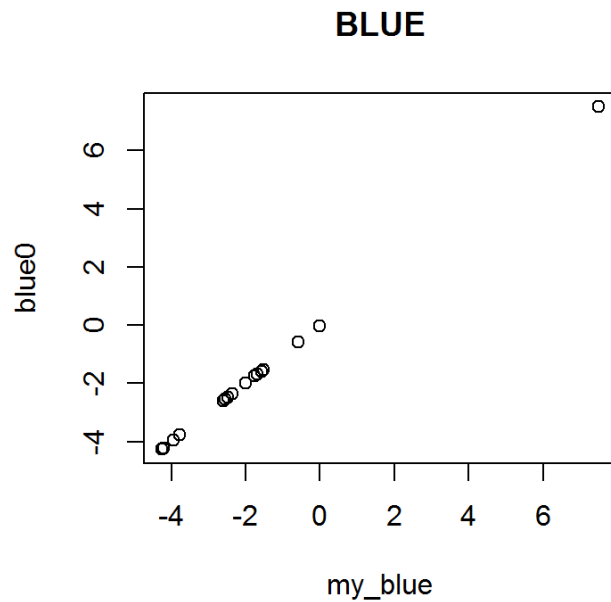
## Example 3 - DIY BLUPs

```
iK = chol2inv(K)
Lambda = 0.6575786/0.2334026
W = cbind(X,Z)
Sigma = as.matrix(Matrix::bdiag(diag(0,ncol(X)),iK*Lambda))
LHS = crossprod(W) + Sigma
RHS = crossprod(W,y)
g = solve(LHS,RHS)
my_blue = g[ c(1:ncol(X))]
my_gebv = g[-c(1:ncol(X))]
```



## Example 3 - DIY BLUPs

```
par(mfrow=c(1,2))  
plot(my_blue,blue0,main="BLUE")  
plot(my_gebv,gebv0,main="GEBV")
```



## Example 3 - Check variance components

$$\sigma_e^2 = \frac{e'y}{n-p} \text{ and } \sigma_u^2 = \frac{u'K^{-1}u + \text{tr}(K^{-1}C^{22}\sigma_e^2)}{q}$$

```
e = y - X %*% my_blue - Z %*% my_blup
```

```
Ve = c(y%*%e)/(length(y)-ncol(X))
```

```
Ve
```

```
## [1] 0.6330582
```

```
trKC22 = sum(diag(iK%*(solve(LHS)[(ncol(X)+1):(ncol(W)),(ncol(X)+1):(ncol(W))])))
```

```
Vg = c(t(my_blup)%*%iK%*my_blup+trKC22*Ve)/ncol(Z)
```

```
Vg
```

```
## [1] 0.1569456
```

# Example 4 - Soybeans

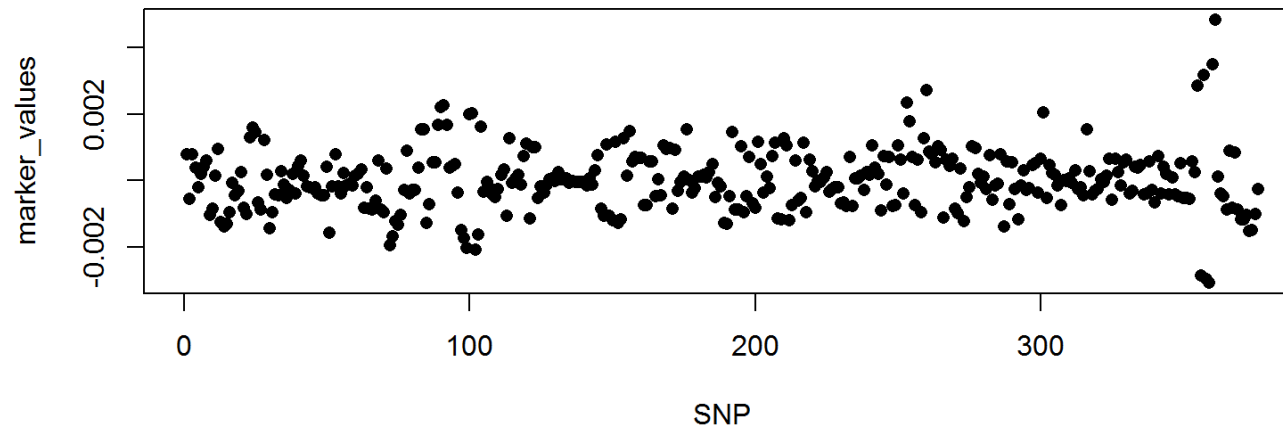
# snp-BLUP

```
data(tpod,package='NAM')  
X = matrix(1,length(y),1)  
Z = gen  
dim(Z)
```

```
## [1] 196 376
```

## Example 3 - Fit the model

```
# Fit using the lme4 package  
fit0 = reml(y=y,X=X,Z=Z) # same as reml(y=y,Z=gen)  
marker_values = fit0$EBV  
gebv0 = c(gen %*% marker_values)  
# Marker effects  
plot(marker_values,pch=16, xlab='SNP')
```

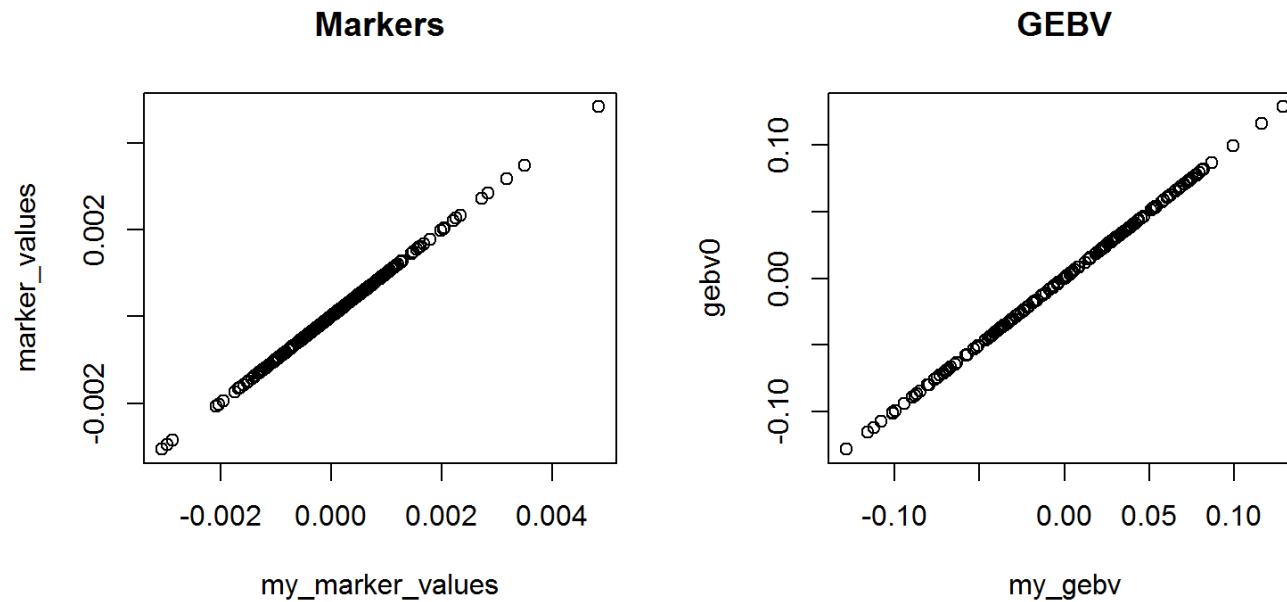


## Example 3 - DIY BLUPs

```
iK = diag(ncol(Z))
Lambda = fit0$VC[2] / fit0$VC[1]
W = cbind(X,Z)
Sigma = diag( c(0,rep(Lambda,ncol(Z))) )
LHS = crossprod(W) + Sigma
RHS = crossprod(W,y)
g = solve(LHS,RHS)
my_mu = g[ c(1:ncol(X))]
my_marker_values = g[-c(1:ncol(X))]
my_gebv = c(gen %*% my_marker_values) # GEBVs from RR
```

## Example 3 - DIY BLUPs

```
par(mfrow=c(1,2))  
plot(my_marker_values, marker_values, main="Markers")  
plot(my_gebv, gebv0, main="GEBV")
```



# Example 3 - Heritability from RR

```
fit0$VC
```

```
##           Vg           Ve           h2
## 1 1.659819e-05 0.03167014 0.0005238214
```

```
Scale=sum(apply(gen,2,var)); Va=fit0$VC[1]*Scale; Ve=fit0$VC[2]
round((Va/(Va+Ve)),2)
```

```
##      Vg
## 1 0.16
```

```
K = tcrossprod(apply(gen,2,function(x) x-mean(X)))
K = K/mean(diag(K)); round(reml(y,K=K)$VC,2)
```

```
##      Vg      Ve      h2
## 1 0.01 0.03 0.16
```



# Estimate VC from bad starters

```

W = cbind(X,Z); iK = diag(ncol(Z))
Ve = Vg = 1 # Bad starting values
for(i in 1:100){ # Check the VC convergence after few iterations
  Lambda = Ve/Vg;
  Sigma = as.matrix(Matrix::bdiag(diag(0,ncol(X)),iK*Lambda))
  LHS = crossprod(W) + Sigma; RHS = crossprod(W,y); g = solve(LHS,RHS)
  my_blue = g[ c(1:ncol(X))]; my_blup = g[-c(1:ncol(X))]
  e = y - X%%my_blue - Z%%my_blup; Ve = c(y%%e)/(length(y)-ncol(X))
  trKC22 = sum(diag(iK%%(solve(LHS)[(ncol(X)+1):(ncol(W)),(ncol(X)+1):(ncol(W))]])))
  Vg = c(t(my_blup)%%iK%%my_blup+trKC22*Ve)/ncol(Z)
  if(!i%%25){cat('It',i,'VC:  Vg =',Vg,'and Ve =',Ve,'\n')}}

## It 25 VC:  Vg = 0.0002960602 and Ve = 0.01801226
## It 50 VC:  Vg = 7.428217e-05 and Ve = 0.02531553
## It 75 VC:  Vg = 4.000965e-05 and Ve = 0.02818575
## It 100 VC:  Vg = 2.887763e-05 and Ve = 0.02956763

```