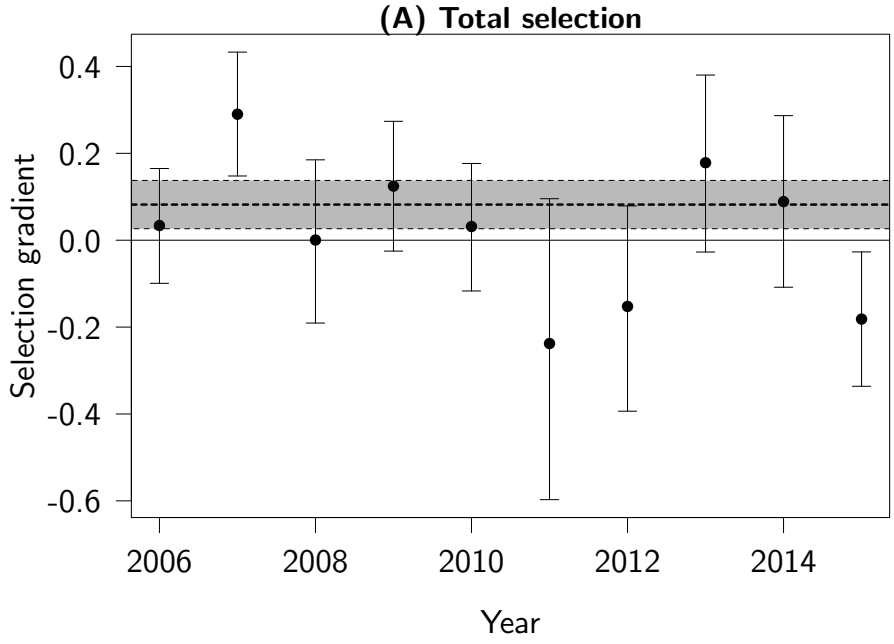
```
## Loading required package:  Matrix
## Loading required package:  coda
## Loading required package:  ape
```

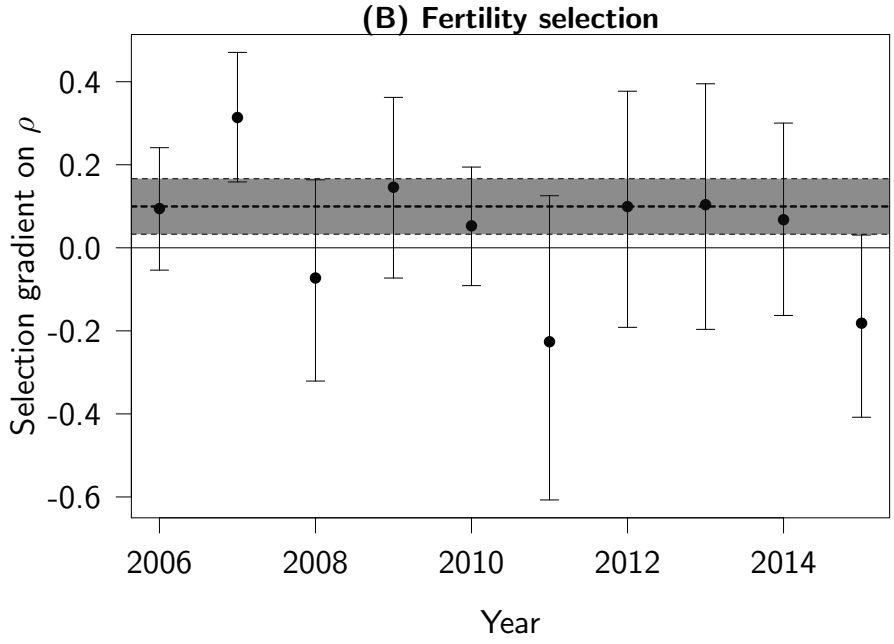
```
setPar()
plot(SelAByYear, x=2006:2015, ylim=c(min( CISelAByYear), max( CISelAByYear)), xlab="Year", y
abline(h=0)
arrows(x0 = 2006:2015,x1 = 2006:2015,code = 3, y0 = CISelAByYear[1,],
       y1 = CISelAByYear[2,], angle = 90,length = 0.1)
abline(h=coefficients(m0all)[2], lty=2, lwd=5)
lowm0all <- coefficients(m0all)[2]+1.96*sm0all$coefficients[2,2]
highm0all <- coefficients(m0all)[2]-1.96*sm0all$coefficients[2,2]
polygon(x=c(2005,2016,2016,2005),y=c(lowm0all,lowm0all, highm0all, highm0all),
        fillOddEven = TRUE, col=rgb(0.1,0.1,0.1,0.3), lty=2)
```



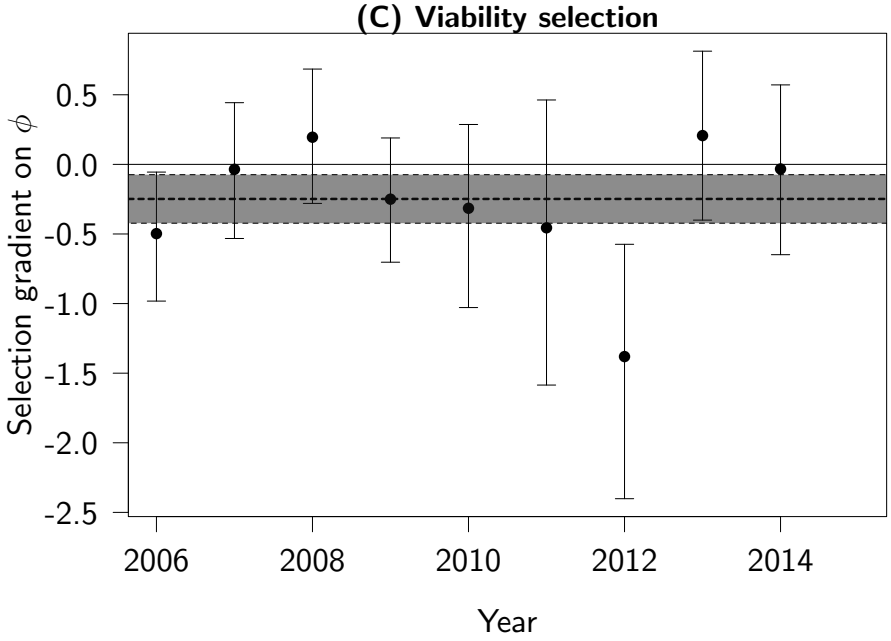

```
#points(x=2006:2015,y=unlist(coefficients(mmRnoCorfitness)$Year["StMass"]), pch=17)
```

```
setPar()
plot(SelAByYearRho, x=2006:2015, ylim=c(min( CISelAByYearRho), max( CISelAByYearRho)), xlab=
abline(h=0)
arrows(x0 = 2006:2015,x1 = 2006:2015,code = 3, y0 = CISelAByYearRho[1,],
       y1 = CISelAByYearRho[2,], angle = 90,length = 0.1)
```

```
abline(h=coefficients(m0allRho)[2], lty=2, lwd=5)
sm0allRho <- summary(m0allRho)
lowm0allRho <- coefficients(m0allRho)[2]+1.96*sm0allRho$coefficients[2,2]
highm0allRho <- coefficients(m0allRho)[2]-1.96*sm0allRho$coefficients[2,2]
polygon(x=c(2005,2016,2016,2005),y=c(lowm0allRho,lowm0allRho, highm0allRho, highm0allRho),
        fillOddEven = TRUE, col=rgb(0.1,0.1,0.1,0.5), lty=2)
```




```
setPar()
plot(SelAByYearPhi, x=2006:2015, ylim=c(min( CISelAByYearPhi, na.rm=TRUE), max( CISelAByYear
abline(h=0)
arrows(x0 = 2006:2015,x1 = 2006:2015,code = 3, y0 = CISelAByYearPhi[1,],
       y1 = CISelAByYearPhi[2,], angle = 90,length = 0.1)
abline(h=coefficients(m0allphi)[2], lty=2, lwd=5)
lowm0allphi <- coefficients(m0allphi)[2]+1.96*sm0allphi$coefficients[2,2]
highm0allphi <- coefficients(m0allphi)[2]-1.96*sm0allphi$coefficients[2,2]
polygon(x=c(2005,2016,2016,2005),y=c(lowm0allphi,lowm0allphi, highm0allphi, highm0allphi),
        fillOddEven = TRUE, col=rgb(0.1,0.1,0.1,0.5), lty=2 )
```



Correlation fertility viability

```
cor.test(YearPheno$Phi,YearPheno$Rho)

##
##  Pearson's product-moment correlation
##
## data:  YearPheno$Phi and YearPheno$Rho
## t = -1.9473, df = 1292, p-value = 0.05171
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1082724891  0.0003989614
## sample estimates:
##         cor
## -0.05409695
```

```
rounding <- 3

BetaGlm<- c(paste(round(sm0all$coefficients[2,1],rounding)," (",round(sm0all$coefficients[2,
            paste(round(sm0allRho$coefficients[2,1],rounding)," (",round(sm0allRho$coefficie
            paste(round(sm0allphi$coefficients[2,1],rounding)," (",round(sm0allphi$coefficie

SDyears <- c(sd(SelAByYear),sd(SelAByYearRho),sd(SelAByYearPhi, na.rm=TRUE))
SEyears <- c(mean(SeSelAByYear), mean(SeSelAByYearRho),mean(SeSelAByYearPhi,na.rm=T))
BetaGLMM <- c(paste(round(smmARnoCorfitness$coefficients[2,1],rounding)," (",round(smmARnoCo
```

```
            paste(round(smmRnoCorrho$coefficients[2,1],rounding)," (",round(smmRnoCorrho$coe
            paste(round(smmRnoCorphi$coefficients[2,1],rounding)," (",round(smmRnoCorphi$coe
SigmaA <- c(sqrt(as.numeric(smmARnoCorfitness$varcor$Year.1)),
            sqrt(as.numeric(smmRnoCorrho$varcor$Year.1)),
sqrt(as.numeric(smmRnoCorphi$varcor$Year.1)))
SigRat <- c(sqrt(as.numeric(smmARnoCorfitness$varcor$Year.1))/smmARnoCorfitness$coefficients
            sqrt(as.numeric(smmRnoCorrho$varcor$Year.1))/smmRnoCorrho$coefficients[2,1],
sqrt(as.numeric(smmRnoCorphi$varcor$Year.1))/smmRnoCorphi$coefficients[2,1])

psigmaA <- c(fitnessAanova$`Pr(>Chisq)`[2]/2, RhoAanova$`Pr(>Chisq)`[2]/2, PhiAanova$`Pr(>Ch
confsigma <- c(paste("[",round(CImmARnoCorfitness[2,1],rounding),";",round(CImmARnoCorfitnes
               paste("[",round(CImmRnoCorrho[2,1],rounding),";",round(CImmRnoCorrho[2,2],rou
               paste("[",round(CImmRnoCorphi[2,1],rounding),";",round(CImmRnoCorphi[2,2],rou

TabSel <- data.frame(BetaGlm = BetaGlm, B=SDyears, C=SEyears , D=BetaGLMM , E=SigmaA, DD =co
```

Table 1:

0.082 (0.028)	0.167	0.097	0.036 (0.044)	0.117	[0.063;0.218]	8.1E-06	3.241
0.1 (0.034)	0.160	0.117	0.052 (0.044)	0.111	[0.053;0.212]	2.5E-04	2.145
-0.248 (0.089)	0.484	0.319	-0.217 (0.098)	0.109	[0;0.425]	3.6E-01	-0.501

Correlation between selection and evolution

```
posterior.mode(as.mcmc(SelToG))

##     var1
## 0.360992

HPDinterval(as.mcmc(SelToG))

##           lower    upper
## var1 -0.3910723 0.635243
## attr(,"Probability")
## [1] 0.95
```

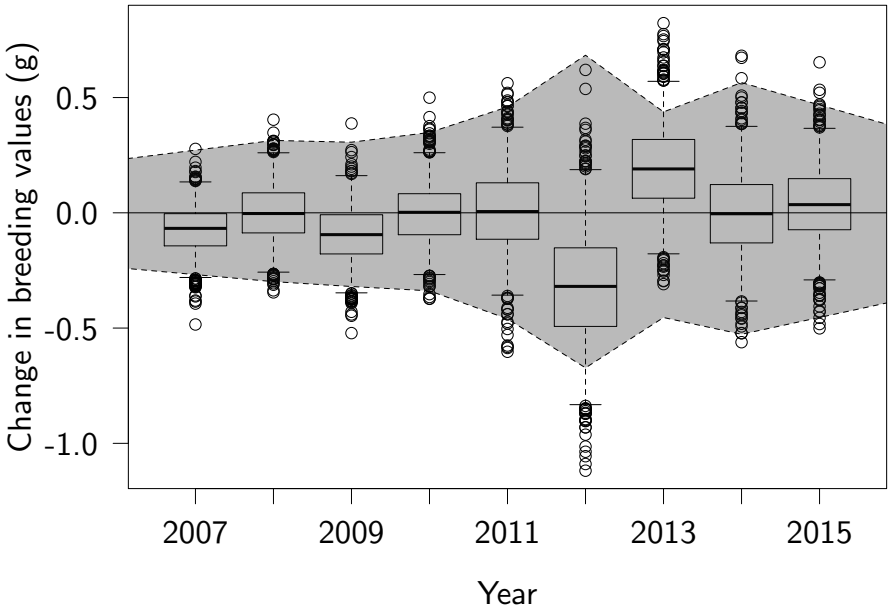
```
szgr <- 2
szax <- 1.3
marr <- c(4, 4, 1, 1) + 0.1
par(las=1,mar=marr, cex=szgr, cex.lab=szax , cex.axis=szax, lwd=2 , las=1)

bbv <- boxplot(bvpairwise,ylab="Change in breeding values (g)", xlab="Year", range = 1,cex=1
polygon(x = c(2006,2008:2014,2016,2016,2014:2008,2006) -2006, y = c(LowDrift,rev(HighDrit)),

bbv$stats
```

```
##             [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] -0.280501568 -0.257383635 -0.347173588 -0.267832000 -0.357132486
## [2,] -0.143366410 -0.087176846 -0.178386364 -0.095028840 -0.114692786
## [3,] -0.067520536 -0.003132208 -0.094707610  0.002214962  0.005447066
## [4,] -0.003739138  0.086721131 -0.008337006  0.082918931  0.130138872
## [5,]  0.134012211  0.260359089  0.161244353  0.260692898  0.371425991
##            [,6]        [,7]         [,8]        [,9]
## [1,] -0.8321504 -0.17794910 -0.382623528 -0.29096950
## [2,] -0.4928537  0.06339068 -0.130565493 -0.07342106
## [3,] -0.3192183  0.19033674 -0.004056911  0.03549705
## [4,] -0.1519085  0.31816724  0.122591670  0.14822783
## [5,]  0.1872504  0.57031481  0.374995899  0.36644027
```

```
bbv$group
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [106] 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [141] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [176] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [211] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7
## [246] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8
## [281] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9
## [316] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [351] 9
```

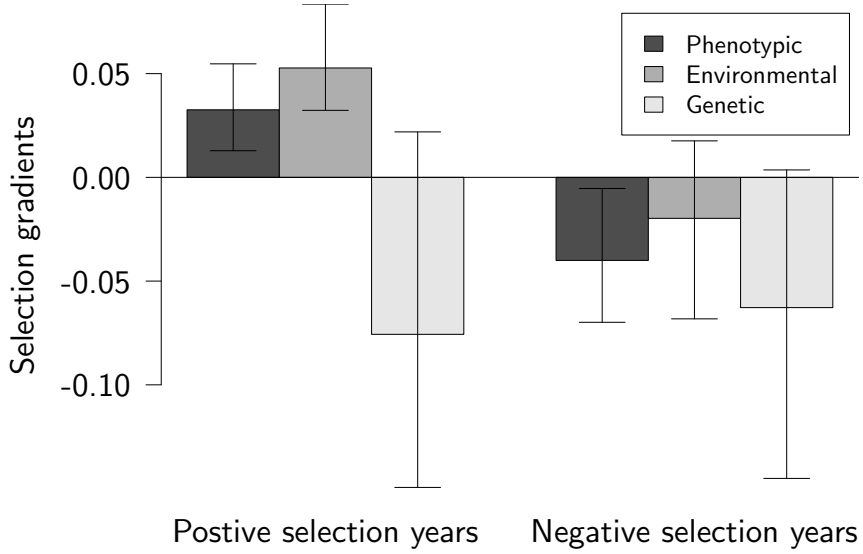
```
abline(h=0)
```




```
density(bvpairwise[,1])

##
## Call:
##  density.default(x = bvpairwise[, 1])
##
## Data: bvpairwise[, 1] (1000 obs.); Bandwidth 'bw' = 0.02343
##
##        x                 y
##  Min.   :-0.5547   Min.   :0.000192
##  1st Qu.:-0.3292   1st Qu.:0.018199
##  Median :-0.1036   Median :0.424183
##  Mean   :-0.1036   Mean   :1.107278
##  3rd Qu.: 0.1219   3rd Qu.:2.109751
##  Max.   : 0.3475   Max.   :3.900320
```

```
setPar()
par(mar=c(4, 6, 1, 1) + 0.1)
Betas <- matrix(sapply(X = list(BetaP1, BetaE1, BetaG1, BetaP2, BetaE2, BetaG2), posterior.m
BetasCI <- sapply(X = list(BetaP1, BetaE1, BetaG1, BetaP2, BetaE2, BetaG2), HPDinterval)

x <- barplot(Betas, beside=TRUE, ylim=c(min(BetasCI),max(BetasCI)),names.arg = c("Postive se
abline(h=0)
arrows(x0 = x, y0=BetasCI[1,],y1=BetasCI[2,],angle = 90,code = 3)
```

```
mtext(side=2, "Selection gradients", line=4, las=0, cex=szax*szgr)
```




```
posterior.mode(BetaG1 - BetaE1)

##       var1
## -0.1234186

HPDinterval(BetaG1 - BetaE1)

##           lower       upper
## var1 -0.2180723 -0.02770633
## attr(,"Probability")
## [1] 0.95

mean((BetaG1 - BetaE1)>0)*2

## [1] 0.006

posterior.mode(BetaG2 - BetaE2)

##         var1
## -0.009610168

HPDinterval(BetaG2 - BetaE2)
```

```
##           lower      upper
## var1 -0.1379426 0.05755331
## attr(,"Probability")
## [1] 0.95

mean((BetaG2 - BetaE2)>0)*2

## [1] 0.424

posterior.mode(BetaE1 - BetaE2)

##       var1
## 0.07575993

HPDinterval(BetaE1 - BetaE2)

##           lower     upper
## var1 0.03828845 0.1372658
## attr(,"Probability")
## [1] 0.95

mean((BetaE1 - BetaE2)<0)*2

## [1] 0

posterior.mode(BetaG1 - BetaG2)

##         var1
## -0.003689573

HPDinterval(BetaG1 - BetaG2)

##            lower      upper
## var1 -0.08005378 0.07575714
## attr(,"Probability")
## [1] 0.95

mean((BetaG1 - BetaG2)>0)*2

## [1] 0.908
```