Contents

Principal Component Analysis

> When using PCA for data reduction rather than inferential analysis, assumption of normality is not required (Tabachnick & Fidell 2001)

We will use principal component analysis to reduce the redundancy among the behavioural measurements scored from the open field (OF) and mirror image stimulation (MIS) trials and to identify the dominant axes of behavioural variation in the OF and MIS trials. Principal components are calculated separately for the OF and MIS behavioural measurements using a correlation matrix. The behavioural dataset used in this analysis is exactly the same as used in Taylor et al. (2012) so the principal component loadings and scores are also the same.

To evaluate the appropriateness of this analyses we will follow Budaev's advice (2010. Using Principal Components and Factor Analysis in Animal Behaviour Research: Caveats and Guidelines. Ethology 116: 472–480.). Budaev suggests some best practices for reporting PCA results that we will follow.

```
library(MASS) # MASS clashes with dplyr... so always load first
library(pander) # pander clashes with dplyr... so always load first
```

```
##
## Attaching package: 'pander'
##
## The following object is masked from 'package:knitr':
##
##     pandoc
```

```
library(foreach)
```

```
## foreach: simple, scalable parallel programming from Revolution Analytics
## Use Revolution R for scalability, fault tolerance and more.
## http://www.revolutionanalytics.com
```

```
library(doMC)
```

```
## Loading required package: iterators
## Loading required package: parallel
```

```
registerDoMC()
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:MASS':
##
##     select
##
## The following objects are masked from 'package:stats':
```

```
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.alignment('right', row.names = 'left')
library(mvnormtest)
library(psych)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
##
## The following object is masked from 'package:psych':
##
##     %+%
```

```
library(MCMCglmm)
```

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

```
behav_data <- tbl_df(read.table(file = "data/behaviour.csv",
                                sep = ',',
                                header = TRUE,
                                stringsAsFactors = FALSE))
behav_data
```

```
## Source: local data frame [4,286 x 25]
##
##     ID Sex Grid Year julian         trial_id Obs docil handlevent_year
## 1    4   F   SU 2005    177               NA MRG    28              12
## 2  601   M   AG 2005    165               NA CLS    15               9
## 3  601   M   AG 2005    182 0.44270.2005.182 ADI    17              10
## 4  601   M   AG 2005    224 0.44270.2005.224 ADI     8              12
## 5    5   F   KL 2005    170 0.46255.2005.170 ADI    10              10
## 6    5   F   KL 2005    184               NA MAW    20              11
## 7    5   F   KL 2005    212 0.46255.2005.212 ADI    12              14
## 8    5   F   KL 2005    219               NA ADI    10              15
## 9  603   M   AG 2005    170               NA CLS    17               7
## 10 603   M   AG 2005    173 0.46342.2005.173 ADI    12               8
## .. ... ... ...  ...    ...              ... ...   ...             ...
## Variables not shown: Study (chr), front (dbl), attack_rate (dbl), back
##   (dbl), ln_attack_latency (dbl), ln_approach_latency (dbl), hole_rate
##   (dbl), jump_rate (dbl), chew (dbl), still (dbl), hang (dbl), groom
##   (dbl), walk (dbl), fecal (dbl), trial_life (int), trial_year (int)
```

Mirror Image Stimulation PCA

Budaev suggests using the Bartlett's test and the Kaiser–Meyer–Olkin (KMO) measure to assess sampling adequecy. Because the behaviour data contains multiple measures per individual we will first subsample the data, randomly choosing 1 record per individual. Bootstrap 100 times.

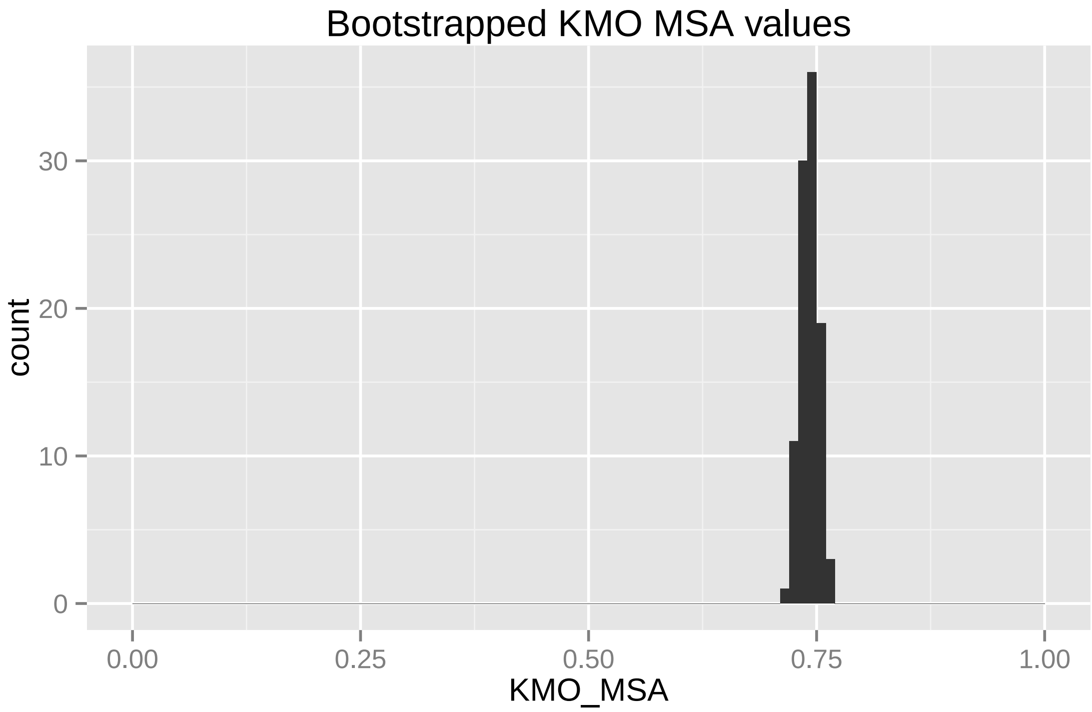
```
mis_data <-  behav_data %>%
  select(ID, trial_id, front, attack_rate, back, ln_attack_latency,
    ln_approach_latency)%>%
  filter(!is.na(front))

# Get only complete records of the MIS behaviours
mis_sub_data <- foreach(i = 1:100, .combine = 'rbind') %dopar% {
  mis_data %>%
  group_by(ID) %>%
  do(sample_n(., 1)) %>%
  mutate(itt = i)
}
save(mis_data, mis_sub_data, file = "data/analyses_data/mis_sub.Rdata")
```

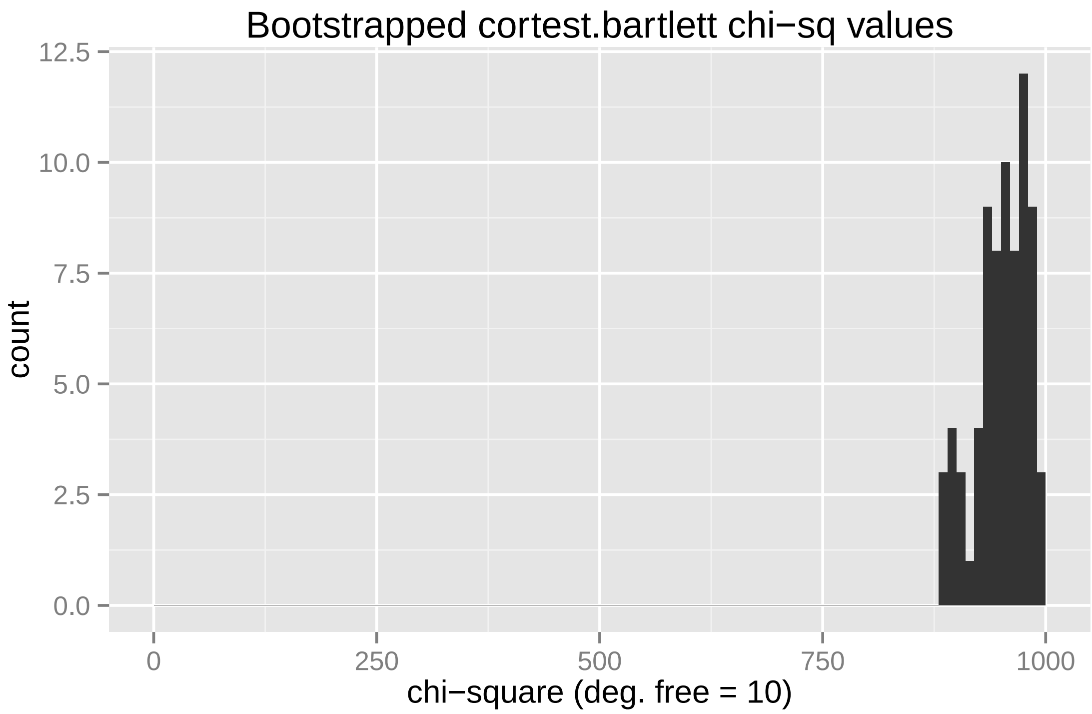
Bartlett's test & KMO

```
load("data/analyses_data/mis_sub.Rdata")
n_trials <- mis_data %>% ungroup() %>% summarise(n = n())
mis_KMO_Bart <- mis_sub_data %>%
    group_by(itt) %>%
    select(-ID, -trial_id) %>%
    summarise(
      KMO_MSA = KMO(cbind(front, attack_rate, back,
        ln_attack_latency, ln_approach_latency))$MSA,
      cortest.bartlett = cortest.bartlett(R = cor(cbind(front,
        attack_rate, back, ln_attack_latency,
        ln_approach_latency)), n = n_trials$n)$chisq
      )

p <- ggplot(mis_KMO_Bart, aes(x = KMO_MSA))
p <- p + geom_histogram(binwidth = 0.01)
p + ggtitle("Bootstrapped KMO MSA values") + xlim(c(0,1))

## Warning: position_stack requires constant width: output may be incorrect
```

```
p <- ggplot(mis_KMO_Bart, aes(x = cortest.bartlett))
p <- p + geom_histogram(binwidth = 10)
p <- p + ggtitle("Bootstrapped cortest.bartlett chi-sq values")
p + xlim(c(0,1000)) + xlab("chi-square (deg. free = 10)")
```

The overall measure of sampling adequecy is fine (Measure of Sampling Adequacy = 0.7415). Bartletts test unsurprisingly rejects the hypotheses that all correlations are zero (P = 0).


Bootstrapped KMO MSA values
count
30
20
10
0
0.00
0.25
0.50
0.75
1.00
KMO_MSA


Bootstrapped cortest.bartlett chi−sq values
count
12.5
10.0
7.5
5.0
2.5
0.0
0
250
500
750
1000
chi−square (deg. free = 10)

Multivariate normality

```
load("data/analyses_data/mis_sub.Rdata")
mis_one <- mis_sub_data %>% filter(itt == 1)
mshapiro.test(t(as.matrix(mis_one[-c(1,2,8)])))
```

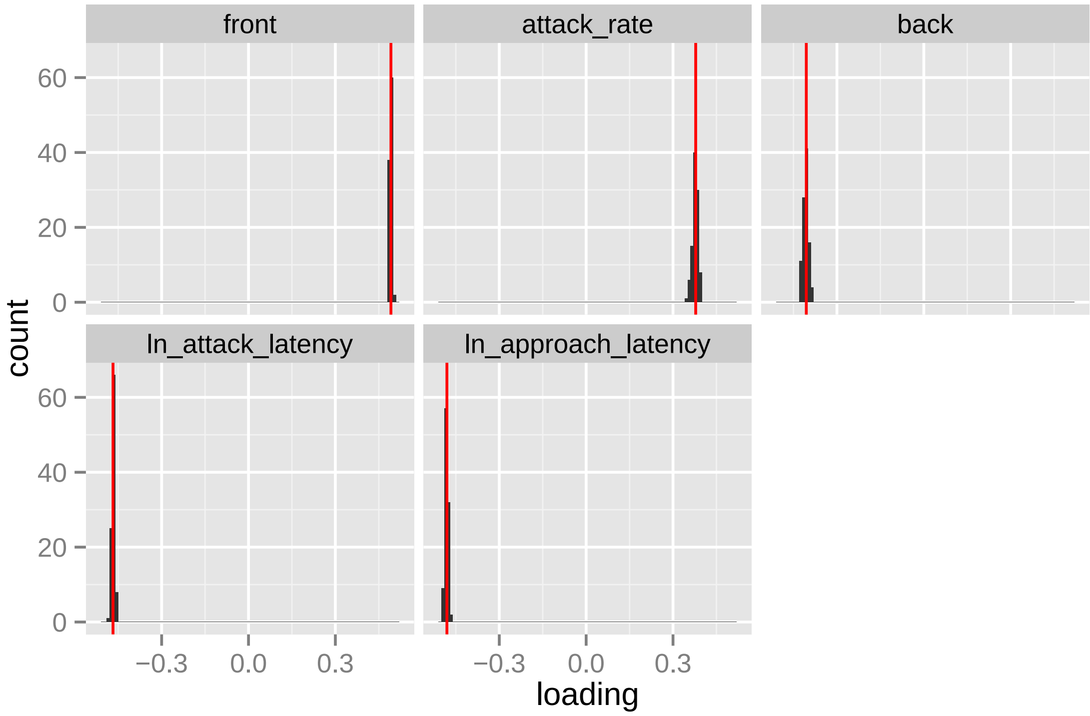
```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.5897, p-value < 2.2e-16
```

The data are not multi-normal. In our case this isn't a major problem because we are not performing any statistical tests alongside the PCA, we are just using the PCA to reduce the dimensionality of the data. We will note that the data are not multivariate normal.

Pseudo-replication

Because our data contains multiple records per individual we will check to be sure pesudo-replication isn't having a large effect on the PCA loadings.

```
load("data/analyses_data/mis_sub.Rdata")
# Calculate PCA for subsampled data
mis_pca <- foreach(i = 1:100, .combine = rbind) %do% {
  foo <- mis_sub_data %>% filter(itt == i)
  pc_loadings <- prcomp(foo[-c(1,2,8)], scale = TRUE)$rotation[ ,"PC1"]
  if(sign(pc_loadings["front"]) == -1) {pc_loadings <- pc_loadings * -1}
  pc_loadings
}

mis_pca <- gather(data.frame(mis_pca), front, attack_rate, back, ln_attack_latency, ln_approach_late

# Calculate PCA for full dataset
mis_pca_full <- prcomp(mis_data[-c(1,2, 8)], scale = TRUE)

mis_pca_loadings <- data.frame(trait = dimnames(mis_pca_full$rotation)[[1]],
                               loading = mis_pca_full$rotation[ ,"PC1"])

if(sign(mis_pca_loadings$loading[mis_pca_loadings$trait == "front"]) == -1) {
  mis_pca_loadings$loading <- mis_pca_loadings$loading * -1
}

p <- ggplot(mis_pca, aes(x = loading))
p <- p + geom_histogram(binwidth = 0.01) + facet_wrap( ~ trait)
p + geom_vline(data = mis_pca_loadings,
      aes(xintercept = loading), color = 'red')
```



The loadings for PC 1 are nearly identical (subsampled to 1 trial per indidivual vs. full dataset). So we will continue with the loadings from the full dataset so that they are consistent with Taylor et al. 2012.

```
#  Reverse sign if 'front' is negative. This way higher scores will always be
#  more aggressive. Sign of pc scores is arbitrary. Make sure to reverse scores
#  and loadings or else confusion!
if(mis_pca_full$rotation["front", "PC1"] < 0){
    mis_pca_full$rotation <- -1 * (mis_pca_full$rotation)
    mis_scores <- data.frame(-mis_pca_full$x)
    } else {
    mis_scores <- data.frame(mis_pca_full$x)
}

# reattach scores to trail.id (which was rowname after PCA)
mis_pca_scores <- data.frame(trial_id = mis_data$trial_id,
                   misPC1 = mis_scores$PC1,
                   misPC2 = mis_scores$PC2,
                   stringsAsFactors = FALSE)

mis_pca_summary <- rbind(mis_pca_full$rotation,
  StdDev = mis_pca_full$sdev,
  PropVar = mis_pca_full$sdev^2 / sum(mis_pca_full$sdev^2))
```

Open Field Arena PCA

Following same procedure as above, now for the open field behavioural measures.

```r
# Get only complete records of the MIS behaviours
of_data <- behav_data %>%
             select(ID, trial_id, hole_rate,jump_rate, chew, still, hang,
              groom, walk, fecal) %>%
             filter(!is.na(hole_rate))

of_sub_data <- foreach(i = 1:100, .combine = 'rbind') %dopar% {
  of_data %>%
  group_by(ID) %>%
  do(sample_n(., 1)) %>%
  mutate(itt = i)
}
save(of_data, of_sub_data, file = "data/analyses_data/of_sub_data.RData")
```

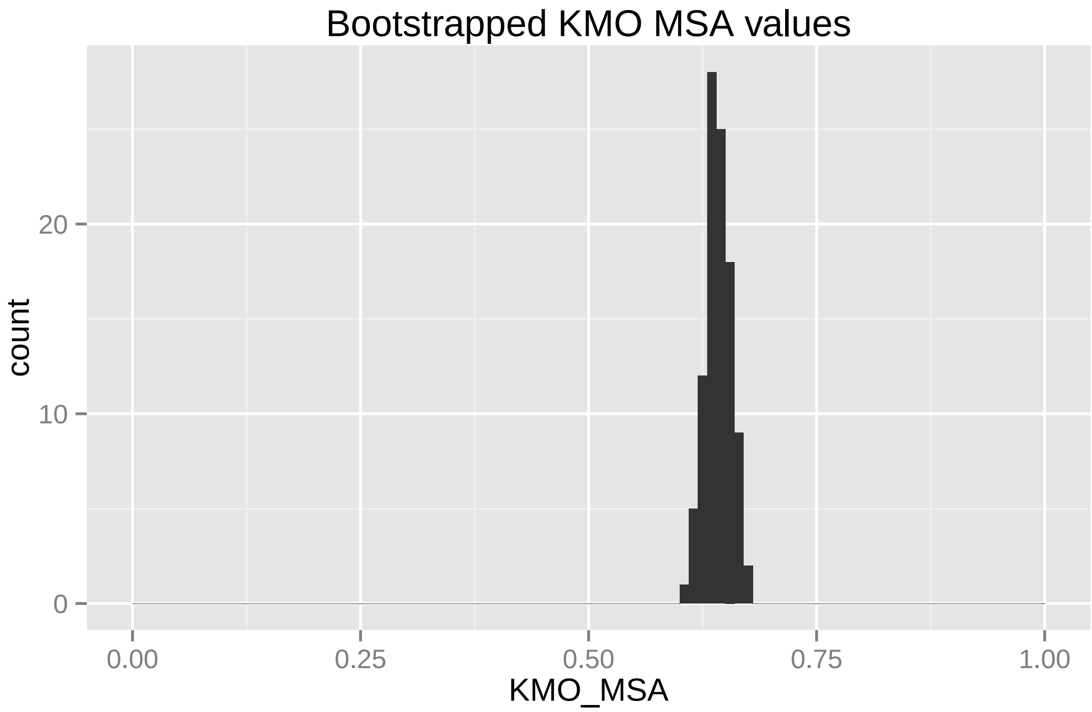
Bartlett's test & KMO

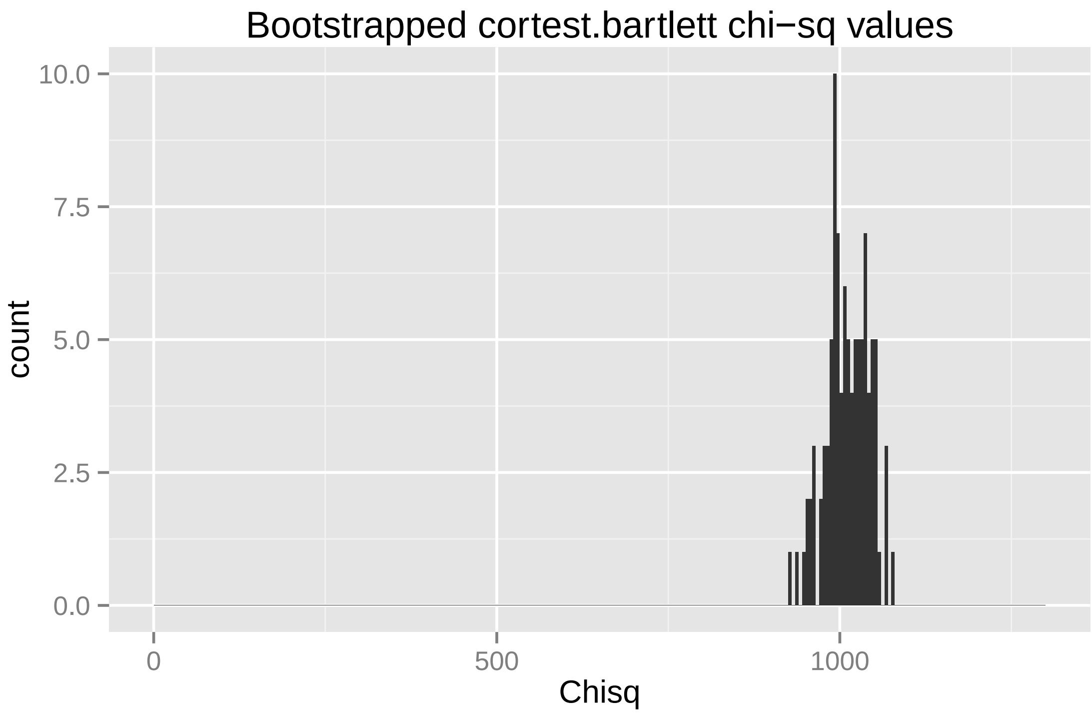
```r
load("data/analyses_data/of_sub_data.RData")
n_trials <- of_data %>% summarise(n = n())
of_KMO_Bart <- of_sub_data %>%
                group_by(itt) %>%
                select(-ID, -trial_id) %>%
                summarise(
                  KMO_MSA = KMO(cbind(hole_rate, jump_rate, chew, still, hang,
                    groom, walk, fecal))$MSA,
                  cortest.bartlett = cortest.bartlett(R = cor(cbind(hole_rate,
                    jump_rate, chew, still, hang, groom, walk, fecal)),
                    n = n_trials$n)$chisq
                  )

p <- ggplot(of_KMO_Bart, aes(x = KMO_MSA)) + geom_histogram(binwidth = 0.01)
p + ggtitle("Bootstrapped KMO MSA values") + xlim(c(0,1))

## Warning: position_stack requires constant width: output may be incorrect


p <- ggplot(of_KMO_Bart, aes(x = cortest.bartlett))
p <- p + geom_histogram(binwidth = 5)
p <- p + ggtitle("Bootstrapped cortest.bartlett chi-sq values")
p + xlim(c(0,1300)) + xlab("Chisq")
```

The overall measure of sampling adequecy is a bit low, maybe..., (MSA = 0.6419). Again, Bartlett's test rejects the hypotheses that all correlations are zero (P = 0).

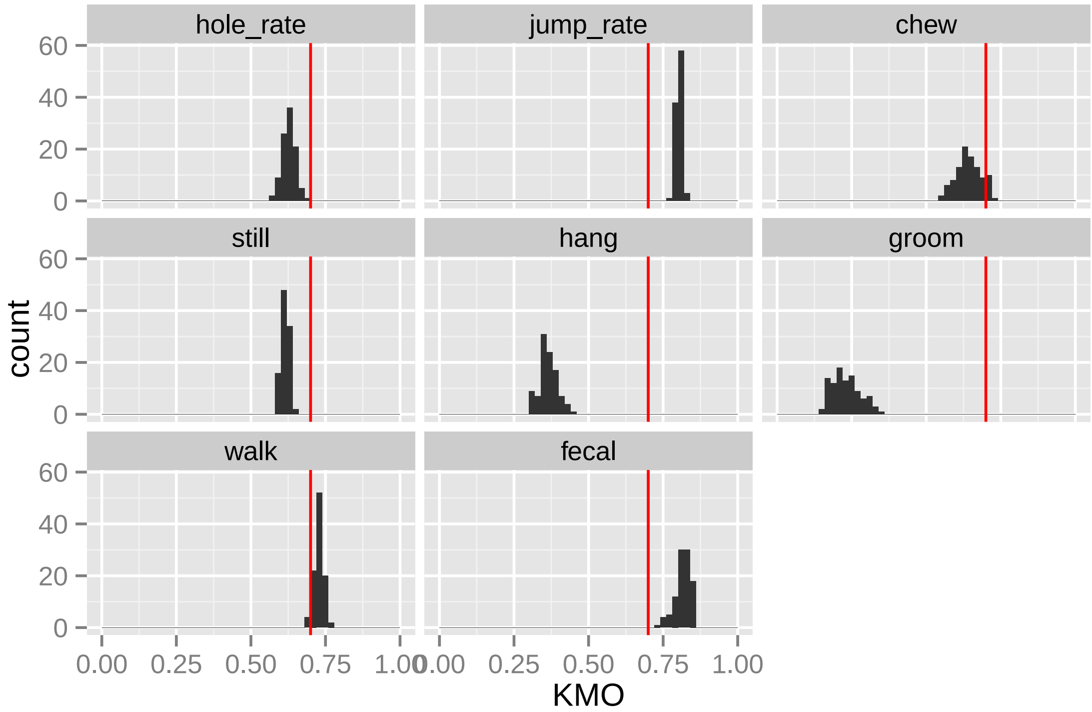
Lets take a closer look at the KMO test.

```r
load("data/analyses_data/of_sub_data.RData")
kmo_of <- foreach(i = 1:100, .combine = 'rbind') %do% {
  foo <- of_sub_data %>% ungroup() %>% filter(itt == i)
  KMO(foo[ ,c(-1,-2,-11)])$MSAi
}
kmo_of <- tbl_df(data.frame(kmo_of))
```


Bootstrapped KMO MSA values
count
20
10
0
0.00
0.25
0.50
0.75
1.00
KMO_MSA


Bootstrapped cortest.bartlett chi−sq values
count
10.0
7.5
5.0
2.5
0.0
0
500
1000
Chisq

```
p <- ggplot(gather(kmo_of, hole_rate, jump_rate, chew, still, hang, groom, walk, fecal, key = "behav
p <- p + facet_wrap( ~ behav) + geom_histogram(binwidth = 0.02) + xlim(c(0,1))
p + geom_vline(xintercept = 0.7, color = 'red')
```



Looks like the low overall KMO index is driven by grooming and hanging. Both of which don't factor in very highly in the PCA loadings. I think therefore this is OK.

Multivariate normality

```
load("data/analyses_data/of_sub_data.RData")
of_one <- of_sub_data %>% filter(itt == 1)
mshapiro.test(t(as.matrix(of_one[-c(1,2,11)])))
```

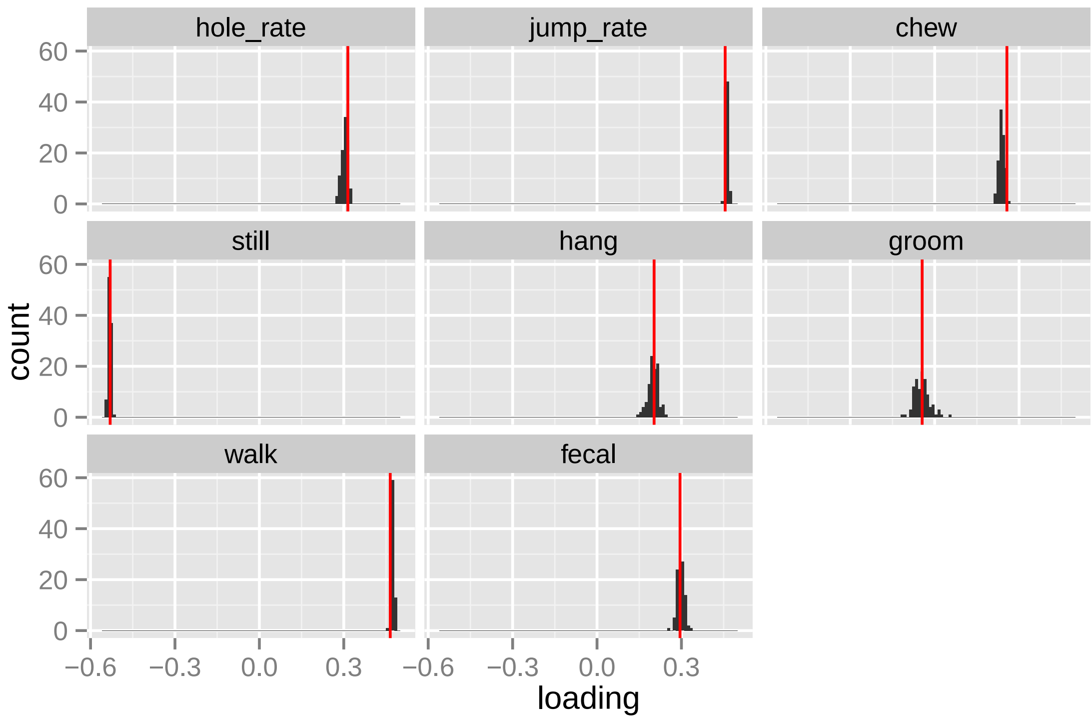
```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.7185, p-value < 2.2e-16
```

The open field data are also not multi-normal (P = 2.4153 × 10-24).

Pseudo-replication

Because our data contains multiple records per individual we will check to be sure pesudo-replication isn't having a large effect on the PCA loadings.

```
load("data/analyses_data/of_sub_data.RData")
# Calculate PCA for subsampled data
of_pca <- foreach(i = 1:100, .combine = rbind) %do% {
  foo <- of_sub_data %>% filter(itt == i)
  pc_loadings <- prcomp(foo[-c(1,2,11)], scale = TRUE)$rotation[ ,"PC1"]
  if(sign(pc_loadings["still"]) == 1) {pc_loadings <- pc_loadings * -1}
}

of_pca <- gather(data.frame(of_pca), hole_rate, jump_rate, chew, still, hang, groom, walk, fecal, ke

# Calculate PCA for full dataset
of_pca_full <- prcomp(of_data[-c(1,2,11)], scale = TRUE)

of_pca_loadings <- data.frame(trait = dimnames(of_pca_full$rotation)[[1]],
                              loading = of_pca_full$rotation[ ,"PC1"])

if(sign(of_pca_loadings$loading[of_pca_loadings$trait == "still"]) == 1) {
  of_pca_loadings$loading <- of_pca_loadings$loading * -1
}

p <- ggplot(of_pca, aes(x = loading)) + geom_histogram(binwidth = 0.01)
p <- p + facet_wrap( ~ trait)
p + geom_vline(data = of_pca_loadings, aes(xintercept = loading), color = 'red')
```



Once again the loadings for PC 1 are spot on the modes of the bootstrap distribution. So we will continue with the loadings from the full dataset so that they are consistent with Taylor et al. 2012. Interestingly, hang and groom have the largest bootstrap variance, and these were the triats identified by the KMO test as not being sampled adequately.

```
load("data/analyses_data/of_sub_data.RData")
of_pca_full <- prcomp(of_data[-c(1,2,11)], scale = TRUE)
#  If the pc coefficient for still is positive, then reverse sign of PC scores
#  so that high PC1 is more active.
if(of_pca_full$rotation["still", "PC1"] > 0){
    of_pca_full$rotation <- -1 * (of_pca_full$rotation)
    of_scores <- data.frame(-of_pca_full$x)
    } else {
    of_scores <- data.frame(of_pca_full$x)
}

#  Get scores
of_pca_scores <- data.frame(trial_id = of_data$trial_id, ofPC1 = of_scores$PC1,
  ofPC2 = of_scores$PC2, ofPC3 = of_scores$PC3, stringsAsFactors = FALSE)

of_pca_summary <- rbind(of_pca_full$rotation, StdDev = of_pca_full$sdev,
  PropVar = of_pca_full$sdev^2 / sum(of_pca_full$sdev^2))

# Save score data
save(mis_pca_summary, of_pca_summary,
  file = "data/analyses_data/pca.RData")
```

Merge PCA data

```
load("data/analyses_data/pca.RData")

# merge pc scores with rest of data by trial id.
pca_data <- left_join(behav_data, mis_pca_scores, by = "trial_id")
pca_data <- left_join(pca_data, of_pca_scores, by = "trial_id")

save(pca_data, mis_pca_summary, of_pca_summary,
  file = "data/analyses_data/pca.RData")

of_table <- c(of_pca_summary["walk",1], of_pca_summary["jump_rate",1],
              of_pca_summary["hole_rate",1], of_pca_summary["fecal",1],
              of_pca_summary["hang",1], of_pca_summary["chew",1],
              of_pca_summary["groom",1], of_pca_summary["still",1],
              of_pca_summary["StdDev",1], of_pca_summary["PropVar",1] * 100)
of_table <- format(of_table, nsmall = 2, digits = 0)

mis_table <- c(mis_pca_summary["front",1], mis_pca_summary["attack_rate",1],
               mis_pca_summary["back",1],
               mis_pca_summary["ln_attack_latency",1],
               mis_pca_summary["ln_approach_latency",1],
               mis_pca_summary["StdDev",1],
               mis_pca_summary["PropVar",1] * 100)
mis_table <-format(mis_table, nsmall = 2, digits = 0)
```

```r
pca_table <- data.frame(check.names = FALSE,
  "OF Behaviour" = c("Walk", "Jump Rate", "Hole Rate", "No. Pellets", "Hang",
  "Chew", "Groom", "Still", "", "Std. Dev.", "% Total variance", "N records",
  "N individuals"),
  "OF PC1" = c(of_table[1:8], "", of_table[9:10], nrow(behav_data %>%
                 filter(!is.na(still))), nrow(of_data %>% select(ID) %>%
                 unique())),
  "MIS Behaviour" = c("Front", "Attack rate", "Back", "Attack latency",
  "Approach latency", rep("", 8)),
  "MIS PC1" = c(mis_table[1:5], rep("", 4), mis_table[6:7],
  nrow(behav_data %>% filter(!is.na(front))), nrow(mis_data %>%
  select(ID) %>% unique()))
  )
pandoc.table(pca_table)
```

OF Behaviour	OF PC1	MIS Behaviour	MIS PC1
Walk	0.47	Front	0.49
Jump Rate	0.46	Attack rate	0.38
Hole Rate	0.31	Back	-0.41
No. Pellets	0.30	Attack latency	-0.47
Hang	0.20	Approach latency	-0.48
Chew	0.26		
Groom	-0.04		
Still	-0.53		
Std. Dev.	1.67		1.67
% Total variance	34.73		55.79
N records	556		553
N individuals	365		364

```r
# sample sizes
n_docil <- pca_data %>% filter(!is.na(docil)) %>%
            group_by(ID) %>% summarise(n = n())

n_oft <- pca_data %>% filter(!is.na(misPC1)) %>%
           group_by(ID) %>% summarise(n = n())

s_table <- data.frame(check.names = FALSE,
      Test = c("OF / MIS", "Handling"),
     "N trials" = c(sum(n_oft$n), sum(n_docil$n)),
     "N individuals" = c(length(unique(n_oft$ID)), length(unique(n_docil$ID))),
     "N > 1 trial"   = c(length(n_oft$ID[n_oft$n > 1]),
                        length(n_docil$ID[n_docil$n > 1]))
```

```
    )
pandoc.table(s_table)
```

Test	N trials	N individuals	N > 1 trial
OF / MIS	553	364	165
Handling	4227	869	621

Random Effects, BLUPs & Repeatability

```
library(MASS) # MASS clashes with dplyr... so always load first
library(pander) # pander clashes with dplyr... so always load first
```

```
##
## Attaching package: 'pander'
##
## The following object is masked from 'package:knitr':
##
##     pandoc
```

```
set.alignment('right', row.names = 'left')
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:MASS':
##
##     select
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(MCMCglmm)
```

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

```
load("data/analyses_data/pca.RData")
```

```
doc_data <- pca_data %>% filter(!is.na(docil))
agg_data <- pca_data %>% filter(!is.na(misPC1))
act_data <- pca_data %>% filter(!is.na(ofPC1))

thin <- 500
burnin <- thin * 100
nitt <- burnin + thin * 1000
```

Docility

Models with covariates for the behavioral tests as fixed effects and ID as a random effect. Covariates include the julian day of the test (continuous), the observer who administered the test (factor) and the handling event number for the year (continuous). We will save 1000 samples from the posterior distribution.

Setting pr = TRUE in MCMCglmm saves the posterior distribution of random effects.

```
prior <- list(
  G = list(G1 = list(V = var(doc_data$docil, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(doc_data$docil, na.rm = TRUE), nu = 0.002)
  )

time_start <- Sys.time()
doc_mcmc_model <- MCMCglmm(docil ~ julian + Obs + handlevent_year +
                             I(handlevent_year^2),
                      random = ~ ID,
                      prior = prior,
                      pr = TRUE,
                      data = doc_data,
                      thin = thin,
                      burnin = burnin,
                      nitt = nitt,
                      verbose = FALSE
                      )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))

## [1] "Approx. model run time:  26.52 mins"

save(doc_mcmc_model, file = "data/analyses_data/doc_mcmc_model.RData")
```

Docility Repeatability

Repeatability is the ratio of the between individual variance to the total variance of the trait (within and between individual variance).

```
load("data/analyses_data/doc_mcmc_model.RData")

PM_HPD <- function(x){
  # Get the posterior mode and HPD interval for a posterior distribution
  out <- posterior.mode(x)
```

```
  out[2:3] <- HPDinterval(x)
  return(out)
}

format_PM_HPD <- function(x){
  fx <- format(x, digits = 2, nsmall = 2)
  out <- fx[[1]]
  out[2] <- paste("(", fx[2], " - ", fx[3], ")", sep = '')
  return(out)
}

doc_I_var <- PM_HPD(doc_mcmc_model$VCV[ ,"ID"])
doc_P_var <- PM_HPD(mcmc(rowSums(doc_mcmc_model$VCV)))
doc_rep <- doc_I_var / doc_P_var

dIv <- format_PM_HPD(doc_I_var)
dPv <- format_PM_HPD(doc_P_var)
dRv <- format_PM_HPD(doc_rep)

doc_table <- data.frame(
   Parameter  = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(dIv[1], dPv[1], dRv[1]),
  "Cred Int." = c(dIv[2], dPv[2], dRv[2])
  )

pandoc.table(doc_table, caption = "Docility repeatability using all trials")
```

Parameter	Post.Mode	Cred.Int.
ID Variance	19.95	(16.63 – 22.25)
Phen. Variance	53.17	(49.79 – 55.76)
Repeatability	0.38	(0.33 – 0.40)

Table 3: Docility repeatability using all trials

Docility Repeatability Across Years

The above model treats repeated measures within a year the same as repeated measures across years (both exist in the dataset). Next we will subset the data to include only across year repeated measures.

```
load("data/analyses_data/doc_mcmc_model.RData")
# Split the dataset into within and across year sets
# We will select one random trial for each squirrel from each year
## Now split into groups of ID & Year and sample one trial at random

doc_data_across <- doc_data %>%
  group_by(ID, Year) %>%
  dplyr:::sample_n.grouped_df(1)
```

```r
# Run model again

prior <- list(
  G = list(G1 = list(V = var(doc_data_across$docil, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(doc_data_across$docil, na.rm = TRUE), nu = 1.002)
  )
prior <- list(
  G = list(G1 = list(V = var(doc_data_across$docil, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(doc_data_across$docil, na.rm = TRUE), nu = 1.002)
)

time_start <- Sys.time()

doc_mcmc_model_across <- MCMCglmm(docil ~ julian + Obs + handlevent_year +
                                    I(handlevent_year^2),
                                  random = ~ ID,
                                  prior = prior,
                                  pr = TRUE,
                                  data = ungroup(doc_data_across),
                                  thin = thin,
                                  burnin = burnin,
                                  nitt = nitt,
                                  verbose = FALSE
                                  )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))

## [1] "Approx. model run time:  10.28 mins"

save(doc_mcmc_model_across,
  file = "data/analyses_data/doc_mcmc_model_across.RData")

load("data/analyses_data/doc_mcmc_model_across.RData")

doc_I_var_across <- PM_HPD(doc_mcmc_model_across$VCV[ ,"ID"])
doc_P_var_across <- PM_HPD(mcmc(rowSums(doc_mcmc_model_across$VCV)))
doc_rep_across   <- doc_I_var_across / doc_P_var_across

dIv_a <- format_PM_HPD(doc_I_var_across)
dPv_a <- format_PM_HPD(doc_P_var_across)
dRv_a <- format_PM_HPD(doc_rep_across)

doc_ay_table <- data.frame(check.names = FALSE,
   Parameter  = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(dIv_a[1], dPv_a[1], dRv_a[1]),
  "Cred Int." = c(dIv_a[2], dPv_a[2], dRv_a[2])
  )

pandoc.table(doc_ay_table, caption = "Docility repeatability across years")
```

Parameter	Post Mode	Cred Int.
ID Variance	16.08	(11.53 – 20.50)
Phen. Variance	50.85	(47.09 – 55.16)
Repeatability	0.32	(0.24 – 0.37)

Table 4: Docility repeatability across years

Ok, the repeatability across years is a little lower than when within year repeated measures are inlcuded. Might be interesing to see what repeatability is within years only.

Docility Repeatabiltiy Within Years

```
# Now pick 1 year for each squirrel, prioritizing years with most measures
pick_year <- function(x){
  if(length(unique(x$Year)) > 1) {
    table_years <- table(x$Year)
    max_years <- which(table_years == max(table_years))
    year <- as.integer(sample(names(max_years), 1))
    x[x$Year == year, ]
    }else{x}
}

doc_data_within <- doc_data %>%
  group_by(ID) %>%
  do(pick_year(.))

# Run model again
prior <- list(
  G = list(G1 = list(V = var(doc_data_within$doc, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(doc_data_within$doc, na.rm = TRUE), nu = 1.002))

## Warning: Name partially matched in data frame
## Warning: Name partially matched in data frame

time_start <- Sys.time()
doc_mcmc_model_within <- MCMCglmm(docil ~ julian + Obs + handlevent_year +
                                   I(handlevent_year^2),
                          random = ~ ID,
                          prior = prior,
                          pr = TRUE,
                          data = ungroup(doc_data_within),
                          thin = thin,
                          burnin = burnin,
                          nitt = nitt,
                          verbose = FALSE
                          )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))
```

```
## [1] "Approx. model run time:  22.08 mins"
```

```
save(doc_mcmc_model_within,
  file = "data/analyses_data/doc_mcmc_model_within.RData")

load("data/analyses_data/doc_mcmc_model_within.RData")

doc_I_var_within <- PM_HPD(doc_mcmc_model_within$VCV[ ,"ID"])
doc_P_var_within <- PM_HPD(mcmc(rowSums(doc_mcmc_model_within$VCV)))
doc_rep_within   <- doc_I_var_within / doc_P_var_within

dIv_w <- format_PM_HPD(doc_I_var_within)
dPv_w <- format_PM_HPD(doc_P_var_within)
dRv_w <- format_PM_HPD(doc_rep_within)

doc_wy_table <- data.frame(check.names = FALSE,
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(dIv_w[1], dPv_w[1], dRv_w[1]),
  "Cred Int." = c(dIv_w[2], dPv_w[2], dRv_w[2])
  )

pandoc.table(doc_wy_table, caption = "Docility repeatability within years",
  justify = 'right')
```

Parameter	Post Mode	Cred Int.
ID Variance	21.20	(17.70 – 23.61)
Phen. Variance	52.45	(49.50 – 55.89)
Repeatability	0.40	(0.36 – 0.42)

Table 5: Docility repeatability within years

Aggression

Aggression is the first principal component of the mirror image stimluation test.

```
prior <- list(
  G = list(G1 = list(V = var(agg_data$misPC1, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(agg_data$misPC1, na.rm = TRUE), nu = 1.002))

time_start <- Sys.time()
agg_mcmc_model <- MCMCglmm(misPC1 ~ julian + trial_life + I(trial_life^2),
                    random = ~ ID,
                    prior = prior,
                    pr = TRUE,
                    data = ungroup(agg_data),
                    thin = thin,
```

```
                    burnin = burnin,
                    nitt = nitt,
                    verbose = FALSE
                    )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))
```

```
## [1] "Approx. model run time:  4.033 mins"
```

```
print(summary(agg_mcmc_model))
```

```
##
##  Iterations = 50001:549501
##  Thinning interval  = 500
##  Sample size  = 1000
##
##  DIC: 2053
##
##  G-structure:  ~ID
##
##    post.mean l-95% CI u-95% CI eff.samp
## ID      1.01    0.588     1.46     1000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units      1.79     1.43     2.13     1000
##
##  Location effects: misPC1 ~ julian + trial_life + I(trial_life^2)
##
##                 post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)       1.03670  0.26825  1.83944     1000  0.018 *
## julian           -0.00736 -0.01143 -0.00367     1000 <0.001 ***
## trial_life        0.35863 -0.43893  1.10061      785  0.358
## I(trial_life^2)  -0.07384 -0.28611  0.11870      762  0.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
save(agg_mcmc_model, file = "data/analyses_data/agg_mcmc_model.RData")
```

Aggression Repeatability

```
load("data/analyses_data/agg_mcmc_model.RData")
agg_I_var <- PM_HPD(agg_mcmc_model$VCV[ ,"ID"])
agg_P_var <- PM_HPD(mcmc(rowSums(agg_mcmc_model$VCV)))
agg_rep <- agg_I_var / agg_P_var

agIv <- format_PM_HPD(agg_I_var)
agPv <- format_PM_HPD(agg_P_var)
agRv <- format_PM_HPD(agg_rep)
```

```r
agg_table <- data.frame(
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(agIv[1], agPv[1], agRv[1]),
  "Cred Int." = c(agIv[2], agPv[2], agRv[2])
  )

pandoc.table(agg_table, caption = "Aggression repeatability using all trials",
  justify = 'right')
```

Parameter	Post.Mode	Cred.Int.
ID Variance	1.04	(0.59 – 1.46)
Phen. Variance	2.66	(2.48 – 3.18)
Repeatability	0.39	(0.24 – 0.46)

Table 6: Aggression repeatability using all trials

Aggression Repeatability Across Years

```r
agg_data_across <- agg_data %>%
  group_by(ID, Year) %>%
  dplyr:::sample_n.grouped_df(1)

prior <- list(
  G = list(
    G1 = list(V = var(agg_data_across$misPC1, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(agg_data_across$misPC1, na.rm = TRUE), nu = 1.002))

time_start <- Sys.time()
agg_mcmc_model_across <- MCMCglmm(misPC1 ~ julian + trial_life +
                                    I(trial_life^2),
                                  random = ~ ID,
                                  prior = prior,
                                  pr = TRUE,
                                  data = ungroup(agg_data_across),
                                  thin = thin,
                                  burnin = burnin,
                                  nitt = nitt,
                                  verbose = FALSE
                                  )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))

## [1] "Approx. model run time:  3.654 mins"

save(agg_mcmc_model_across,
  file = "data/analyses_data/agg_mcmc_model_across.RData")
```

```
load("data/analyses_data/agg_mcmc_model_across.RData")

agg_I_var_across <- PM_HPD(agg_mcmc_model_across$VCV[ ,"ID"])
agg_P_var_across <- PM_HPD(mcmc(rowSums(agg_mcmc_model_across$VCV)))
agg_rep_across   <- agg_I_var_across / agg_P_var_across

agIv_a <- format_PM_HPD(agg_I_var_across)
agPv_a <- format_PM_HPD(agg_P_var_across)
agRv_a <- format_PM_HPD(agg_rep_across)

agg_ay_table <- data.frame(
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(agIv_a[1], agPv_a[1], agRv_a[1]),
  "Cred Int." = c(agIv_a[2], agPv_a[2], agRv_a[2])
  )

pandoc.table(agg_ay_table, caption = "Aggression repeatability across years",
  justify = 'right')
```

Parameter	Post.Mode	Cred.Int.
ID Variance	0.68	(0.41 – 1.40)
Phen. Variance	2.92	(2.54 – 3.30)
Repeatability	0.23	(0.16 – 0.42)

Table 7: Aggression repeatability across years

Aggression Repeatabiltiy Within Years

```
agg_data_within <- agg_data %>%
  group_by(ID) %>%
  do(pick_year(.))

prior <- list(
  G = list(
    G1 = list(V = var(agg_data_within$misPC1, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(agg_data_within$misPC1, na.rm = TRUE), nu = 1.002)
  )

time_start <- Sys.time()
agg_mcmc_model_within <- MCMCglmm(misPC1 ~ julian + trial_life +
                                  I(trial_life^2),
                         random = ~ ID,
                         prior = prior,
                         pr = TRUE,
                         data = ungroup(agg_data_within),
                         thin = thin,
                         burnin = burnin,
```

```r
                            nitt = nitt,
                            verbose = FALSE
                            )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))
```

```
## [1] "Approx. model run time:  3.663 mins"
```

```r
save(agg_mcmc_model_within,
  file = "data/analyses_data/agg_mcmc_model_within.RData")

load("data/analyses_data/agg_mcmc_model_within.RData")

agg_I_var_within <- PM_HPD(agg_mcmc_model_within$VCV[ ,"ID"])
agg_P_var_within <- PM_HPD(mcmc(rowSums(agg_mcmc_model_within$VCV)))
agg_rep_within   <- agg_I_var_within / agg_P_var_within

agIv_w <- format_PM_HPD(agg_I_var_within)
agPv_w <- format_PM_HPD(agg_P_var_within)
agRv_w <- format_PM_HPD(agg_rep_within)

agg_wy_table <- data.frame(
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(agIv_w[1], agPv_w[1], agRv_w[1]),
  "Cred Int." = c(agIv_w[2], agPv_w[2], agRv_w[2])
  )

pandoc.table(agg_wy_table, caption = "Aggression repeatability within years")
```

Parameter	Post.Mode	Cred.Int.
ID Variance	1.39	(0.91 – 1.85)
Phen. Variance	2.78	(2.44 – 3.18)
Repeatability	0.50	(0.37 – 0.58)

Table 8: Aggression repeatability within years

Activity

Activity is the first principal component of the open field test.

```r
prior <- list(
  G = list(G1 = list(V = var(act_data$ofPC1, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(act_data$ofPC1, na.rm = TRUE), nu = 1.002)
  )

time_start <- Sys.time()
act_mcmc_model <- MCMCglmm(ofPC1 ~ julian + trial_life + I(trial_life^2),
```

```r
                    random = ~ ID,
                    prior = prior,
                    pr = TRUE,
                    data = ungroup(act_data),
                    thin = thin,
                    burnin = burnin,
                    nitt = nitt,
                    verbose = FALSE
                    )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))
```

```
## [1] "Approx. model run time:  3.956 mins"
```

```r
print(summary(act_mcmc_model))
```

```
##
##  Iterations = 50001:549501
##  Thinning interval  = 500
##  Sample size  = 1000
##
##  DIC: 1965
##
##  G-structure:  ~ID
##
##    post.mean l-95% CI u-95% CI eff.samp
## ID      1.14     0.81      1.5     1000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units      1.42     1.15      1.7     1000
##
##  Location effects: ofPC1 ~ julian + trial_life + I(trial_life^2)
##
##                  post.mean  l-95% CI  u-95% CI eff.samp  pMCMC
## (Intercept)       1.681676  0.992768  2.483235     1000 <0.001 ***
## julian            0.000679 -0.002925  0.004063     1000  0.682
## trial_life       -1.703131 -2.366306 -0.988700     1000 <0.001 ***
## I(trial_life^2)   0.264116  0.063945  0.428435     1000  0.006 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
save(act_mcmc_model, file = "data/analyses_data/act_mcmc_model.RData")
```

Activity Repeatability

```r
load("data/analyses_data/act_mcmc_model.RData")
act_I_var <- PM_HPD(act_mcmc_model$VCV[ ,"ID"])
act_P_var <- PM_HPD(mcmc(rowSums(act_mcmc_model$VCV)))
act_rep <- act_I_var / act_P_var
```

```
acIv <- format_PM_HPD(act_I_var)
acPv <- format_PM_HPD(act_P_var)
acRv <- format_PM_HPD(act_rep)

act_table <- data.frame(
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(acIv[1], acPv[1], acRv[1]),
  "Cred Int." = c(acIv[2], acPv[2], acRv[2])
  )

pandoc.table(act_table, caption = "Activity repeatability using all trials")
```

Parameter	Post.Mode	Cred.Int.
ID Variance	1.20	(0.81 – 1.50)
Phen. Variance	2.52	(2.29 – 2.90)
Repeatability	0.48	(0.35 – 0.52)

Table 9: Activity repeatability using all trials

Activity Repeatability Across Years

```
act_data_across <- act_data %>%
  group_by(ID, Year) %>%
  dplyr:::sample_n.grouped_df(1)

prior <- list(
  G = list(G1 = list(V = var(act_data_across$ofPC1, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(act_data_across$ofPC1, na.rm = TRUE), nu = 1.002))

time_start <- Sys.time()
act_mcmc_model_across <- MCMCglmm(ofPC1 ~ julian + trial_life +
                                  I(trial_life^2),
                           random = ~ ID,
                           prior = prior,
                           pr = TRUE,
                           data = ungroup(act_data_across),
                           thin = thin,
                           burnin = burnin,
                           nitt = nitt,
                           verbose = FALSE
                           )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))

## [1] "Approx. model run time:  3.431 mins"
```

```r
save(act_mcmc_model_across,
  file = "data/analyses_data/act_mcmc_model_across.RData")

load("data/analyses_data/act_mcmc_model_across.RData")

act_I_var_across <- PM_HPD(act_mcmc_model_across$VCV[ ,"ID"])
act_P_var_across <- PM_HPD(mcmc(rowSums(act_mcmc_model_across$VCV)))
act_rep_across   <- act_I_var_across / act_P_var_across

acIv_a <- format_PM_HPD(act_I_var_across)
acPv_a <- format_PM_HPD(act_P_var_across)
acRv_a <- format_PM_HPD(act_rep_across)

act_ay_table <- data.frame(
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(acIv_a[1], acPv_a[1], acRv_a[1]),
  "Cred Int." = c(acIv_a[2], acPv_a[2], acRv_a[2])
  )

pandoc.table(act_ay_table, caption = "Activity repeatability across years")
```

Parameter	Post.Mode	Cred.Int.
ID Variance	0.89	(0.43 – 1.34)
Phen. Variance	2.70	(2.27 – 2.96)
Repeatability	0.33	(0.19 – 0.45)

Table 10: Activity repeatability across years

Activity Repeatabiltiy Within Years

```r
act_data_within <- act_data %>%
  group_by(ID) %>%
  do(pick_year(.))

prior <- list(
  G = list(G1 = list(V = var(act_data_within$ofPC1, na.rm = TRUE), nu = 1.002)),
  R = list(V = var(act_data_within$ofPC1, na.rm = TRUE), nu = 1.002)
  )

time_start <- Sys.time()
act_mcmc_model_within <- MCMCglmm(ofPC1 ~ julian + trial_life +
                                    I(trial_life^2),
                          random = ~ ID,
                          prior = prior,
                          pr = TRUE,
                          data = ungroup(act_data_within),
                          thin = thin,
```

```
                                burnin = burnin,
                                nitt = nitt,
                                verbose = FALSE
                                )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))
```

```
## [1] "Approx. model run time:  3.567 mins"
```

```
save(act_mcmc_model_within,
  file = "data/analyses_data/act_mcmc_model_within.RData")
```

```
load("data/analyses_data/act_mcmc_model_within.RData")

act_I_var_within <- PM_HPD(act_mcmc_model_within$VCV[ ,"ID"])
act_P_var_within <- PM_HPD(mcmc(rowSums(act_mcmc_model_within$VCV)))
act_rep_within   <- act_I_var_within / act_P_var_within

acIv_w <- format_PM_HPD(act_I_var_within)
acPv_w <- format_PM_HPD(act_P_var_within)
acRv_w <- format_PM_HPD(act_rep_within)

act_wy_table <- data.frame(
  Parameter = c("ID Variance", "Phen. Variance", "Repeatability"),
  "Post Mode" = c(acIv_w[1], acPv_w[1], acRv_w[1]),
  "Cred Int." = c(acIv_w[2], acPv_w[2], acRv_w[2])
  )

pandoc.table(act_wy_table, caption = "Activity repeatability within years")
```

Parameter	Post.Mode	Cred.Int.
ID Variance	1.33	(0.97 – 1.81)
Phen. Variance	2.45	(2.25 – 2.95)
Repeatability	0.54	(0.43 – 0.62)

Table 11: Activity repeatability within years

Repeatability Summary

```
d <- function(x){paste(x[1], x[2])}
table_summary <- data.frame(
  Behaviour = c("Aggression", "Activity", "Docility"),
  All     = c(d(agRv), d(acRv), d(dRv)),
  Across = c(d(agRv_a), d(acRv_a), d(dRv_a)),
  Within = c(d(agRv_w), d(acRv_w), d(dRv_w))
  )
save(table_summary, file = "mcmc_repeatability_summary.RData")
```

```r
pandoc.table(table_summary, caption = "Summary of repeatabilities")
```

Behaviour	All	Across	Within
Aggression	0.39 (0.24 – 0.46)	0.23 (0.16 – 0.42)	0.50 (0.37 – 0.58)
Activity	0.48 (0.35 – 0.52)	0.33 (0.19 – 0.45)	0.54 (0.43 – 0.62)
Docility	0.38 (0.33 – 0.40)	0.32 (0.24 – 0.37)	0.40 (0.36 – 0.42)

Table 12: Summary of repeatabilities (continued below)

Run lmer models

```r
library(lme4)
```

```
## Loading required package: Rcpp
```

Docility

```r
doc_data <- pca_data %>% filter(!is.na(docil))
doc_lmer_model <- lmer(docil ~ as.factor(Year) + julian + Obs +
  handlevent_year + I(handlevent_year^2) + (1 | ID), data = doc_data)
save(doc_lmer_model, file = "data/analyses_data/doc_lmer_model.RData")
```

Aggression

```r
agg_data <- pca_data %>% filter(!is.na(misPC1))
agg_lmer_model <- lmer(misPC1 ~  as.factor(Year) + julian + trial_life +
  I(trial_life^2) + (1 | ID), data = agg_data)
save(agg_lmer_model, file = "data/analyses_data/agg_lmer_model.RData")
```

Activity

```r
act_data <- pca_data %>% filter(!is.na(ofPC1))
act_lmer_model <- lmer(ofPC1 ~  as.factor(Year) + julian + trial_life +
  I(trial_life^2) + (1 | ID), data = act_data)
save(act_lmer_model, file = "data/analyses_data/act_lmer_model.RData")
```

Extract random effects / BLUPs

from MCMCglmm models

```
extractMCMCglmmRanEffs <- function(x, trait_name){
  library(dplyr)
  library(MCMCglmm)
  # Now get the posterior distribution, so we can pass variance in behavioral
  # estimate on to further analyses. 1000 sets of random effects saved.
  sols <- data.frame(x$Sol) ## Get random effects
  sols <- sols[ ,grep("ID", names(sols))] ## Get all the ID columns
  sols <- stack(sols)
  names(sols) <- c(trait_name, "ID")
  sols$itt <- 1:1000 # Just an index for each set of random effects (e.g. each
                     # MCMC sample)
  sols$ID <- gsub("ID\\.", "", sols$ID)
  sols$type <- "raneff"
  tbl_df(sols)
}

load("data/analyses_data/doc_mcmc_model.RData")
doc_mcmc_raneffs <- extractMCMCglmmRanEffs(doc_mcmc_model,
  trait_name = "docility")

load("data/analyses_data/agg_mcmc_model.RData")
agg_mcmc_raneffs <- extractMCMCglmmRanEffs(agg_mcmc_model,
  trait_name = "aggression")

load("data/analyses_data/act_mcmc_model.RData")
act_mcmc_raneffs <- extractMCMCglmmRanEffs(act_mcmc_model,
  trait_name = "activity")

# No dplyr outer join?
mcmc_raneffs <- merge(doc_mcmc_raneffs, agg_mcmc_raneffs,
  by = c("ID", "itt", "type"), all = TRUE)
mcmc_raneffs <- merge(mcmc_raneffs, act_mcmc_raneffs,
  by = c("ID", "itt", "type"), all = TRUE)

mcmc_raneffs <- tbl_df(mcmc_raneffs)

save(mcmc_raneffs, file = "data/analyses_data/MCMCraneffs.RData")
```

from lmer models

```
load("data/analyses_data/doc_lmer_model.RData")
load("data/analyses_data/agg_lmer_model.RData")
load("data/analyses_data/act_lmer_model.RData")

extractLmerBLUPs <- function(x, value){
  require(dplyr)
  require(lme4)
```

```r
  blups <- ranef(x)$ID
  blups$ID <- rownames(blups)
  names(blups) <- c(value, "ID")
  blups$itt <- 0  ## NOTE itt == 0 is BLUPs
  blups$type <- "blup"
  tbl_df(blups)
}

doc_lmer_model.blups <- extractLmerBLUPs(doc_lmer_model, "docility")
agg_lmer_model.blups <- extractLmerBLUPs(agg_lmer_model, "aggression")
act_lmer_model.blups <- extractLmerBLUPs(act_lmer_model, "activity")

# No outer join for dplyr?
lmer_blups <- merge(doc_lmer_model.blups, agg_lmer_model.blups, all = TRUE)
lmer_blups <- merge(lmer_blups, act_lmer_model.blups, all = TRUE)
lmer_blups <- tbl_df(lmer_blups)
save(lmer_blups, file = "data/analyses_data/lmer_blups.RData")
```

rbind MCMC raneffs and BLUPs

```r
load("data/analyses_data/MCMCraneffs.RData")
load("data/analyses_data/doc_lmer_model.RData")
raneffs_blups <- rbind(mcmc_raneffs, lmer_blups)

save(raneffs_blups, file = "data/analyses_data/raneffs_blups.RData")
```

Rescale docility raneffs & BLUPs

The docility BLUPs can be rescaled to the raw docility measurement. Aggression and activity are based on PCA scores and so are unit/scaleless so no need to rescale.

```r
load("data/analyses_data/raneffs_blups.RData")
load("data/analyses_data/doc_mcmc_model.RData")
load("data/analyses_data/doc_lmer_model.RData")
# The intercepts are nearly indentical, as expected, but we'll keep them
# separate for consistency
lmer_model_intercept <- summary(doc_lmer_model)$coefficients["(Intercept)",
                                "Estimate"]
mcmc_model_intercept <-  summary(doc_mcmc_model)$solutions["(Intercept)",
                                "post.mean"]

raneffs_blups$docility[raneffs_blups$type == "raneff"] <-
  raneffs_blups$docility[raneffs_blups$type == "raneff"] + mcmc_model_intercept

raneffs_blups$docility[raneffs_blups$type == "blup"] <-
  raneffs_blups$docility[raneffs_blups$type == "blup"] + lmer_model_intercept

save(raneffs_blups, file = "data/analyses_data/raneffs_blups.RData")
```

Model Diagnostics

Docility

```
load("data/analyses_data/doc_mcmc_model.RData")
autocorr.diag(doc_mcmc_model$VCV)
```

```
##                  ID     units
## Lag 0     1.000000  1.000000
## Lag 500   0.002313 -0.036398
## Lag 2500  0.017386  0.008852
## Lag 5000  0.035184  0.006301
## Lag 25000 0.008448  0.007935
```

```
geweke.diag(doc_mcmc_model$VCV)
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##     ID  units
## 0.4782 0.5114
```

```
heidel.diag(doc_mcmc_model$VCV)
```

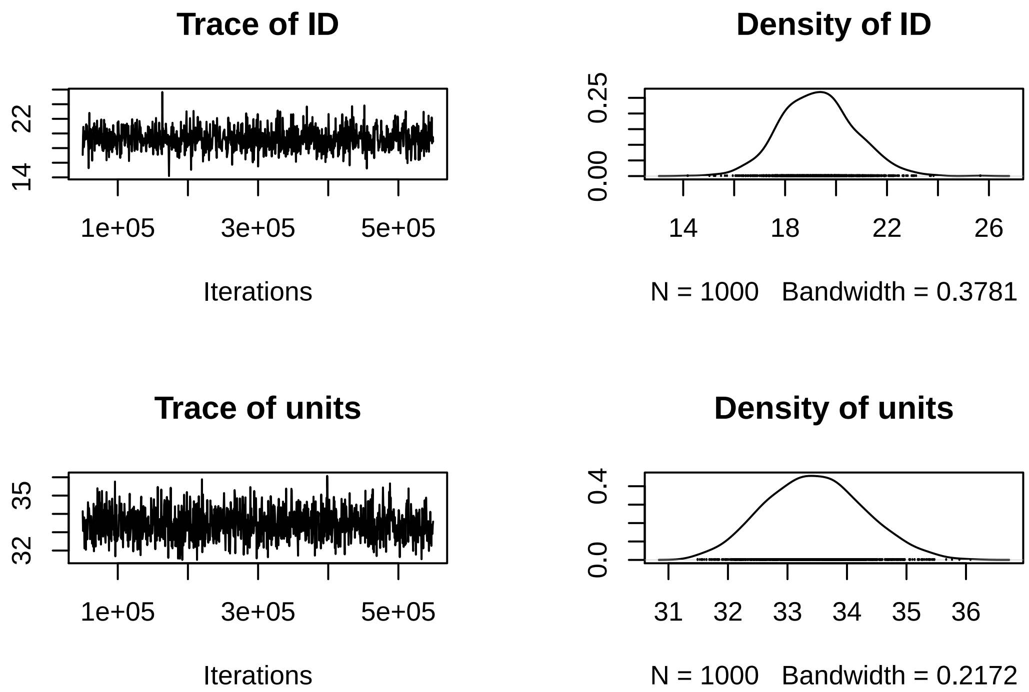
```
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.521
## units passed       1         0.706
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.3 0.0897
## units passed    33.5 0.0506
```

```
plot(doc_mcmc_model$VCV)
```

Aggression

```
load("data/analyses_data/agg_mcmc_model.RData")
autocorr.diag(agg_mcmc_model$VCV)
```

```
##                   ID     units
## Lag 0      1.0000000  1.000000
## Lag 500    0.0192333 -0.004458
## Lag 2500   0.0228034  0.020507
## Lag 5000  -0.0009168 -0.036696
## Lag 25000  0.0020479 -0.024898
```



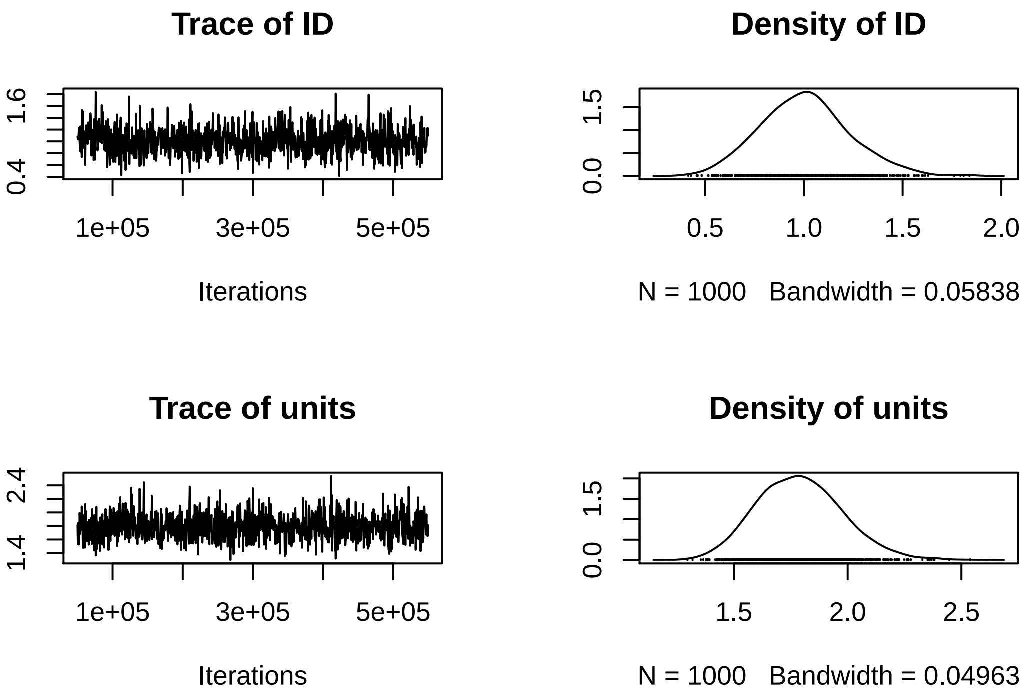

```
geweke.diag(agg_mcmc_model$VCV)
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##     ID  units
##  2.604 -2.004
```

```
heidel.diag(agg_mcmc_model$VCV)
```

```
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.467
## units passed       1         0.938
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    1.01 0.0139
## units passed    1.79 0.0116
```

```
plot(agg_mcmc_model$VCV)
```



Activity

```
load("data/analyses_data/act_mcmc_model.RData")
autocorr.diag(act_mcmc_model$VCV)
```

```
##                   ID       units
## Lag 0        1.00000   1.000000
## Lag 500     -0.01143  -0.001613
## Lag 2500    -0.02268  -0.003608
## Lag 5000     0.03577  -0.017001
## Lag 25000    0.01260   0.003523
```

```
geweke.diag(act_mcmc_model$VCV)
```

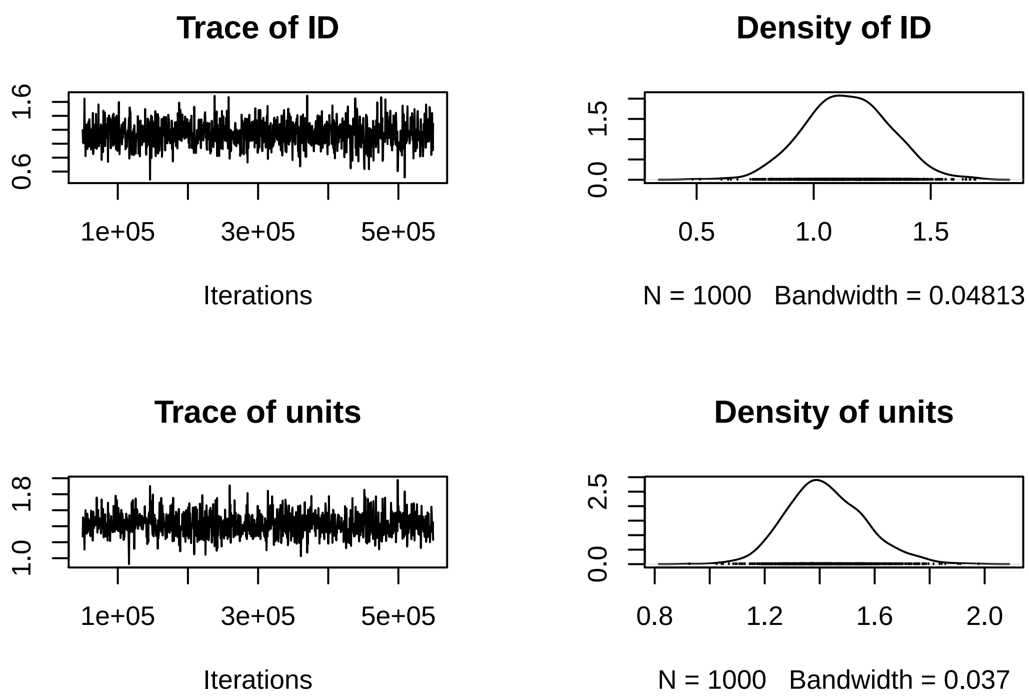
```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##     ID  units
## 0.5648 0.3727
```

```
heidel.diag(act_mcmc_model$VCV)
```

```
##
##        Stationarity start     p-value
```

```
##        test         iteration
## ID     passed       1          0.849
## units passed       1          0.683
##
##        Halfwidth Mean Halfwidth
##        test
## ID     passed     1.14 0.01120
## units passed     1.42 0.00879
```

```
plot(act_mcmc_model$VCV)
```



Assessing the effect of priors on BLUPs

```
load("data/analyses_data/pca.RData")
library(tidyr)
library(dplyr)
library(MCMCglmm)
library(pander)
set.alignment('right', row.names = 'left')

doc_data <- pca_data %>% filter(!is.na(docil))
agg_data <- pca_data %>% filter(!is.na(misPC1))
act_data <- pca_data %>% filter(!is.na(ofPC1))
```

Priors

From a post to r-sig-me by Ned Dochterman

1. Parameter expanded
2. Another parameter expanded just to see if results vary across runs
3. Parameter expanded variance = docility variance
4. Parameter expanded really high variance
5. Inverse Wishart
6. Inverse Gamma
7. Flat, uniform, prior for just a variance
8. Flat improper prior, equivalent to REML fitting.

```
priors <- list(
  list(
    G=list(G1=list(V=1, nu=1, alpha.mu = 0, alpha.V = 10000)),
    R=list(V=1, nu=1)
  ),
  list(
    G=list(G1=list(V=1, nu=1, alpha.mu = 0, alpha.V = 10000)),
    R=list(V=1, nu=1)
  ),
  list(
    G=list(
      G1=list(
        V=var(doc_data$docil, na.rm = TRUE), nu=1, alpha.mu = 0,
          alpha.V = 10000
      )
    ),
    R=list(V=var(doc_data$docil, na.rm = TRUE), nu=1)
  ),
  list(
    G=list(G1=list(V=1000, nu=1, alpha.mu = 0, alpha.V = 1000)),
    R=list(V=1000, nu=1)
  ),
  list(G=list(G1=list(V=1, nu=1)), R=list(V=1, nu=1)),
  list(G=list(G1=list(V=1, nu=0.002)), R=list(V=1, nu=0.002)),
  list(G=list(G1=list(V=1e-16, nu=-2)), R=list(V=1e-16, nu=-2))   ,
  list(G=list(G1=list(V=1,nu=0)),R = list(V =1, nu = 0))
)
```

Run models

```
library(foreach)

## foreach: simple, scalable parallel programming from Revolution Analytics
## Use Revolution R for scalability, fault tolerance and more.
## http://www.revolutionanalytics.com

library(doMC)
```

```
## Loading required package: iterators
## Loading required package: parallel
```

```
registerDoMC(cores = 8)

thin <- 100
burnin <- thin * 100
nitt <- burnin + thin * 1000

time_start <- Sys.time()
m_priors <- foreach(i = 1:length(priors)) %dopar% {
  MCMCglmm(docil ~ julian + Obs + handlevent_year + I(handlevent_year^2),
                                    random = ~ ID,
                                    prior = priors[[i]],
                                    pr = TRUE,
                                    data = doc_data,
                                    thin = thin,
                                    burnin = burnin,
                                    nitt = nitt,
                                    verbose = FALSE
                                    )
}
print(paste("Approx. models run time: ", format(Sys.time() - time_start)))
```

```
## [1] "Approx. models run time:  8.042 mins"
```

```
save(m_priors, file = "data/analyses_data/m_priors.RData")
```

Model Diagnostics

```
load("data/analyses_data/m_priors.RData")

ad <- list()
gd <- list()
hd <- list()

for(i in 1:length(priors)){
  ad[[i]] <- autocorr.diag(m_priors[[i]]$VCV)
  gd[[i]] <- geweke.diag(m_priors[[i]]$VCV)
  hd[[i]] <- heidel.diag(m_priors[[i]]$VCV)
}
ad
```

```
## [[1]]
##                  ID    units
## Lag 0     1.00000  1.00000
## Lag 100  -0.02922 -0.03013
## Lag 500   0.01560  0.02187
## Lag 1000 -0.05264 -0.03655
## Lag 5000  0.01480  0.01472
```

```
##
## [[2]]
##                   ID     units
## Lag 0      1.000000  1.000000
## Lag 100   -0.038620 -0.005688
## Lag 500   -0.038033  0.062510
## Lag 1000   0.041033  0.021035
## Lag 5000   0.008467 -0.030301
##
## [[3]]
##                   ID     units
## Lag 0      1.000000  1.000000
## Lag 100   -0.047543  0.048670
## Lag 500    0.003440 -0.002852
## Lag 1000  -0.005587  0.011396
## Lag 5000   0.003182  0.007828
##
## [[4]]
##                    ID    units
## Lag 0      1.0000000  1.00000
## Lag 100    0.0004055 -0.02710
## Lag 500    0.0480595 -0.03198
## Lag 1000   0.0189893  0.02511
## Lag 5000  -0.0272179  0.01032
##
## [[5]]
##                 ID    units
## Lag 0      1.00000  1.00000
## Lag 100    0.06009 -0.03227
## Lag 500    0.06123  0.02975
## Lag 1000   0.01682 -0.01543
## Lag 5000  -0.05175  0.03395
##
## [[6]]
##                    ID    units
## Lag 0      1.0000000  1.00000
## Lag 100    0.0002831 -0.03890
## Lag 500    0.0032093 -0.02185
## Lag 1000  -0.0009037  0.05010
## Lag 5000  -0.0424693 -0.02452
##
## [[7]]
##                   ID    units
## Lag 0      1.000000  1.00000
## Lag 100    0.025706  0.03615
## Lag 500   -0.018627  0.02782
## Lag 1000   0.007562  0.05121
## Lag 5000  -0.035466 -0.02292
##
## [[8]]
##                 ID     units
```

```
## Lag 0      1.00000  1.000000
## Lag 100  -0.02928 -0.006564
## Lag 500  -0.05204  0.022471
## Lag 1000 -0.01659  0.063554
## Lag 5000 -0.01517 -0.023190
```

```
gd
```

```
## [[1]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##     ID  units
## 0.5828 0.5301
##
##
## [[2]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##       ID    units
## -0.02447 -1.64978
##
##
## [[3]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      ID   units
## -0.4845  2.2787
##
##
## [[4]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    ID units
## 2.160 1.382
##
##
## [[5]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      ID   units
## -0.4190 -0.1041
```

```
##
##
## [[6]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      ID   units
##  0.1051 -1.3288
##
##
## [[7]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##       ID    units
## -0.07847 -2.51430
##
##
## [[8]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      ID   units
## -0.3845 -0.1387
```

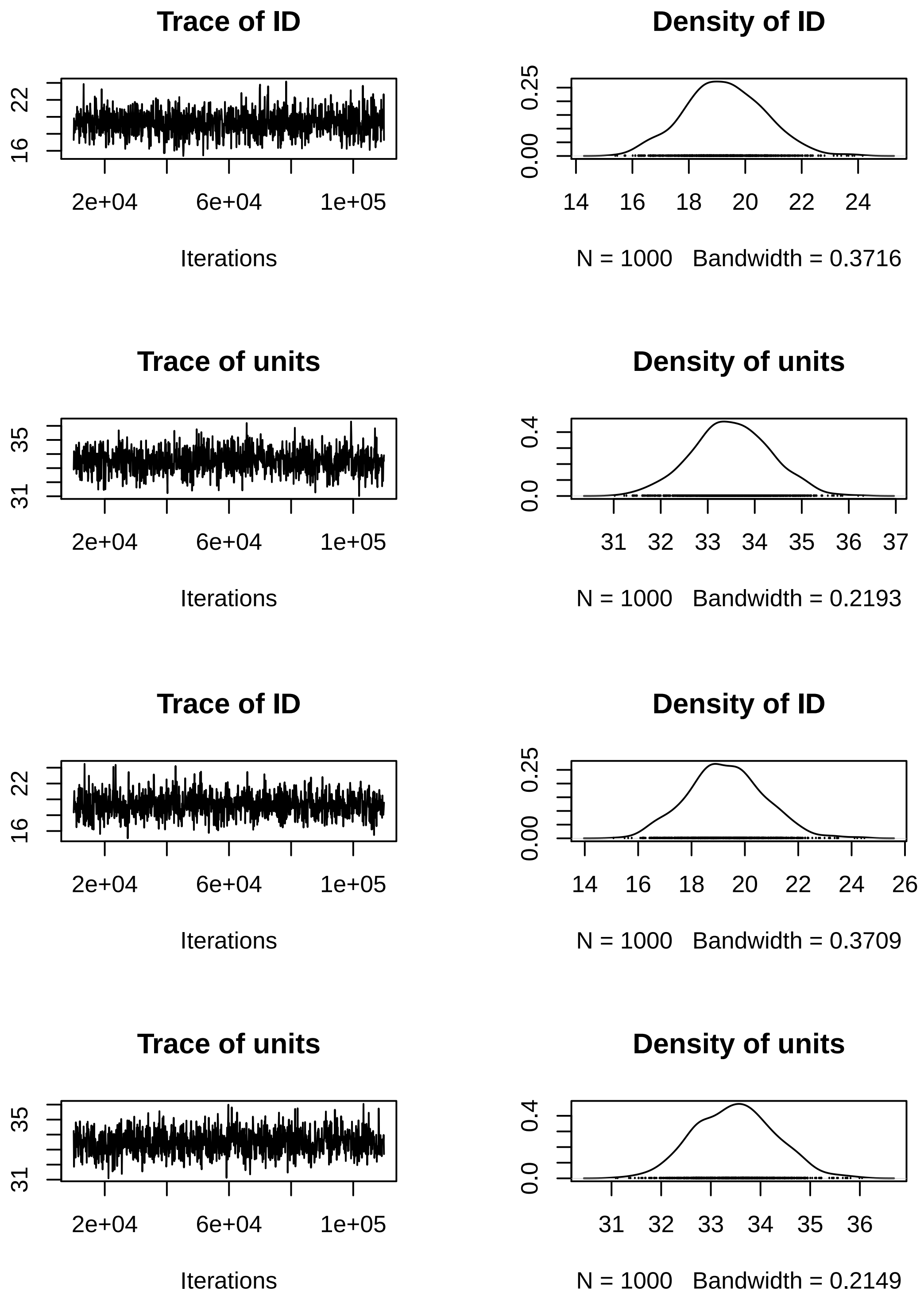
```
hd
```

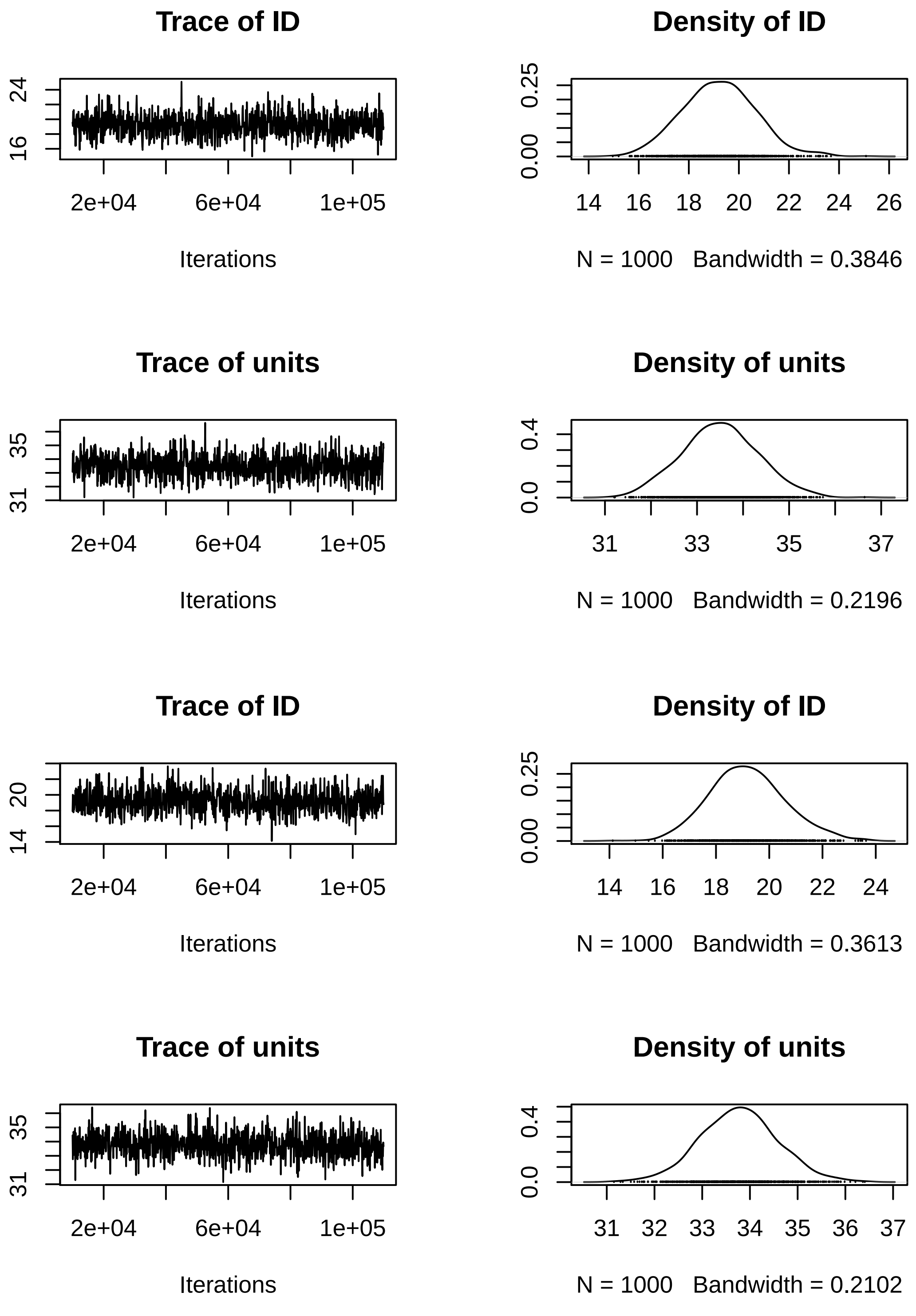
```
## [[1]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.794
## units passed       1         0.321
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.3 0.0880
## units passed    33.5 0.0514
##
## [[2]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.2942
## units passed       1         0.0967
##
##       Halfwidth Mean Halfwidth
##       test
```

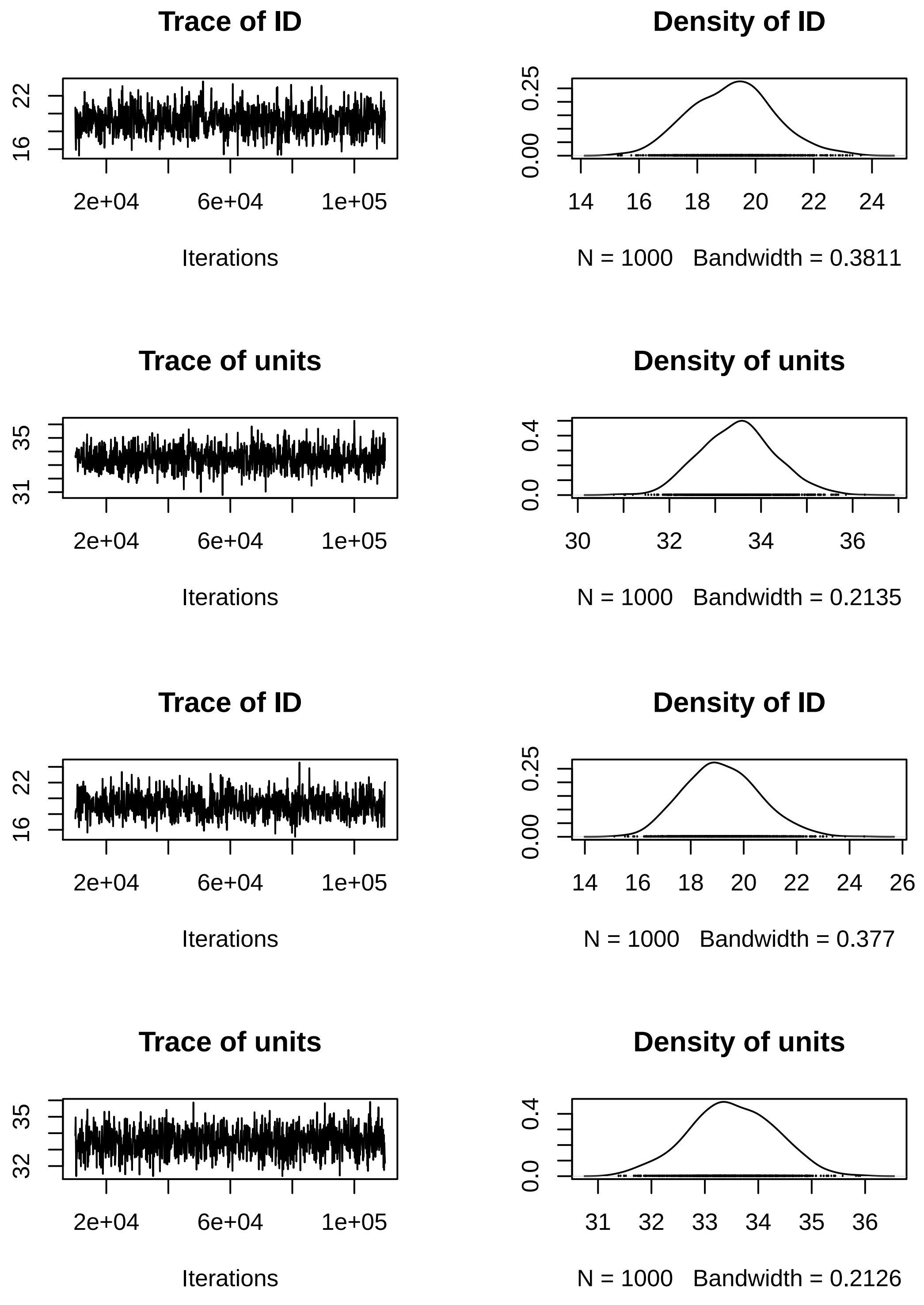
```
## ID    passed    19.3 0.0896
## units passed    33.5 0.0500
##
## [[3]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.800
## units passed       1         0.387
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.2 0.0906
## units passed    33.5 0.0537
##
## [[4]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       101       0.0706
## units passed       301       0.1075
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.2 0.0928
## units passed    33.7 0.0623
##
## [[5]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.572
## units passed       1         0.723
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.2 0.0943
## units passed    33.5 0.0501
##
## [[6]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.278
## units passed       1         0.102
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.2 0.0878
## units passed    33.5 0.0495
##
```

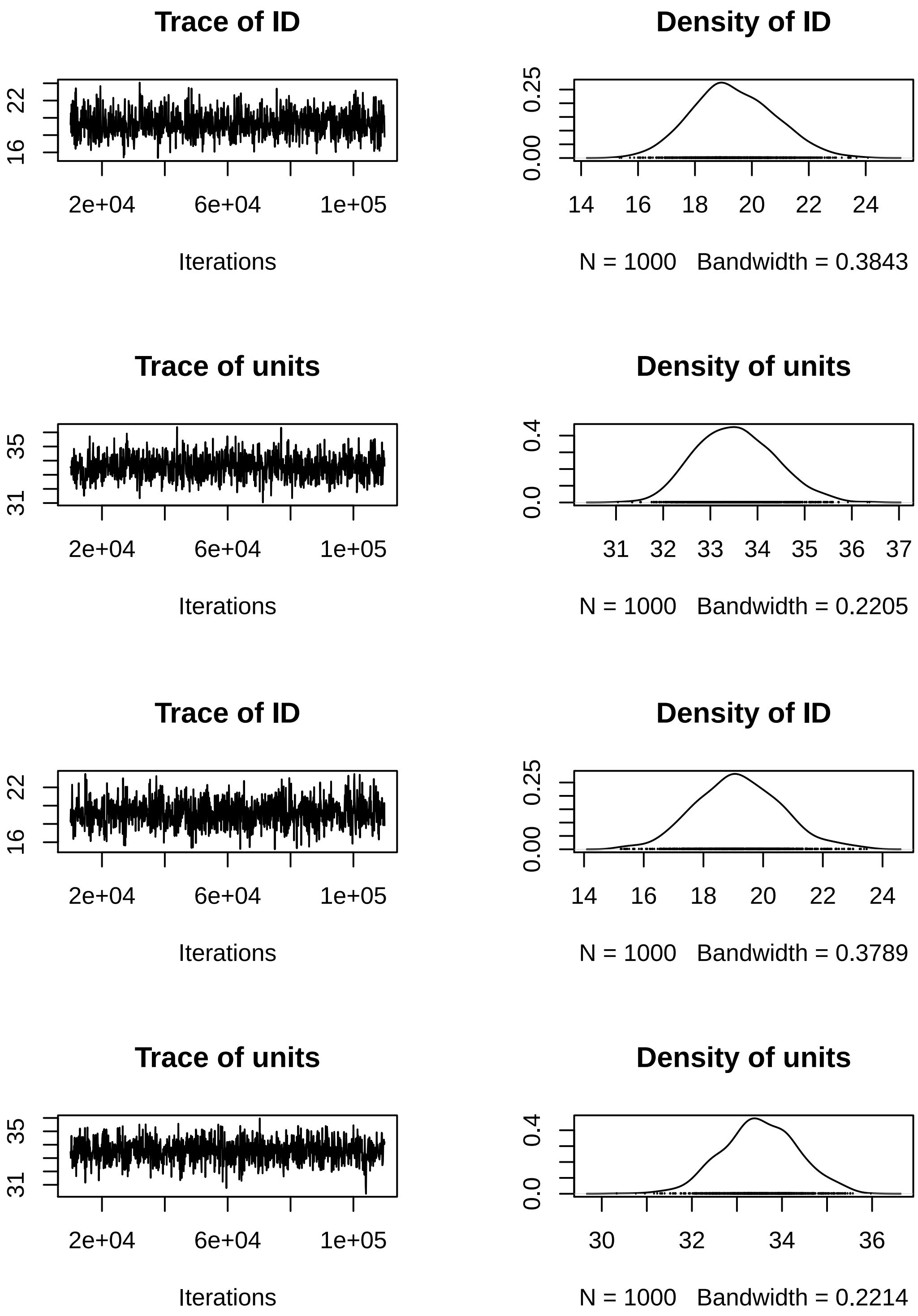
```
## [[7]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.784
## units passed       1         0.487
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.4 0.0895
## units passed    33.5 0.0513
##
## [[8]]
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.535
## units passed       1         0.958
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.2 0.0883
## units passed    33.5 0.0515
```

```
for(i in 1:length(priors)){
  plot(m_priors[[i]]$VCV)
}
```


Trace of ID
22
16
2e+04
6e+04
1e+05
Iterations
Density of ID
0.25
0.00
14
16
18
20
22
24
N = 1000 Bandwidth = 0.3716
Trace of units
35
31
2e+04
6e+04
1e+05
Iterations
Density of units
0.4
0.0
31
32
33
34
35
36
37
N = 1000 Bandwidth = 0.2193
Trace of ID
22
16
2e+04
6e+04
1e+05
Iterations
Density of ID
0.25
0.00
14
16
18
20
22
24
26
N = 1000 Bandwidth = 0.3709
Trace of units
35
31
2e+04
6e+04
1e+05
Iterations
Density of units
0.4
0.0
31
32
33
34
35
36
N = 1000 Bandwidth = 0.2149


Trace of ID
24
16
2e+04
6e+04
1e+05
Iterations
Density of ID
0.25
0.00
14
16
18
20
22
24
26
N = 1000   Bandwidth = 0.3846
Trace of units
35
31
2e+04
6e+04
1e+05
Iterations
Density of units
0.4
0.0
31
33
35
37
N = 1000   Bandwidth = 0.2196
Trace of ID
20
14
2e+04
6e+04
1e+05
Iterations
Density of ID
0.25
0.00
14
16
18
20
22
24
N = 1000   Bandwidth = 0.3613
Trace of units
35
31
2e+04
6e+04
1e+05
Iterations
Density of units
0.4
0.0
31
32
33
34
35
36
37
N = 1000   Bandwidth = 0.2102


Trace of ID
Density of ID
Iterations
N = 1000 Bandwidth = 0.3811
Trace of units
Density of units
Iterations
N = 1000 Bandwidth = 0.2135
Trace of ID
Density of ID
Iterations
N = 1000 Bandwidth = 0.377
Trace of units
Density of units
Iterations
N = 1000 Bandwidth = 0.2126


Trace of ID
22
16
2e+04
6e+04
1e+05
Iterations
Density of ID
0.25
0.00
14
16
18
20
22
24
N = 1000 Bandwidth = 0.3843
Trace of units
35
31
2e+04
6e+04
1e+05
Iterations
Density of units
0.4
0.0
31
32
33
34
35
36
37
N = 1000 Bandwidth = 0.2205
Trace of ID
22
16
2e+04
6e+04
1e+05
Iterations
Density of ID
0.25
0.00
14
16
18
20
22
24
N = 1000 Bandwidth = 0.3789
Trace of units
35
31
2e+04
6e+04
1e+05
Iterations
Density of units
0.4
0.0
30
32
34
36
N = 1000 Bandwidth = 0.2214

Extract raneffs

```
extractMCMCglmmBLUPs <- function(x, value, ptype = "1"){
  p_modes <- posterior.mode(x$Sol) ## Get posterior_modes of the BLUPs
  p_modes <- p_modes[grep("ID", names(p_modes))] ## Get all the ID rows
  p_modes <- stack(p_modes)
  names(p_modes) <- c(value, "ID")
  p_modes$type <- paste("mcmc.mode", ptype, sep = '.')
  p_modes$ID <- gsub("ID\\.", "", p_modes$ID)
  p_modes$itt <- NA
  sols <- data.frame(x$Sol) ## Get BLUPs
  sols <- sols[ ,grep("ID", names(sols))] ## Get all the ID columns
  sols <- stack(sols)
  names(sols) <- c(value, "ID")
  sols$itt <- 1:1000 ## Just an index for each MCMC sample
  sols$type = paste("mcmc", ptype, sep = '.')
  sols$ID <- gsub("ID\\.", "", sols$ID)
  rbind(sols, p_modes)
}

doc_mcmc <- list()
for(i in 1:length(priors)){
  doc_mcmc[[i]] <- extractMCMCglmmBLUPs(m_priors[[i]],
    value = "docility", ptype = i)
}

mcmc_priors <- do.call("rbind", doc_mcmc)
```

Compare MCMC priors

Comparing the effect of priors on the posterior distributions.

Posterior modes

```
mcmc_modes <- mcmc_priors[grep("mode", mcmc_priors$type), ]
mcmc_modes$itt <- NULL
mcmc_modes <- spread(mcmc_modes, type, docility)

cov_modes <- cov(mcmc_modes[ ,2:ncol(mcmc_modes)])
cor_modes <- cor(mcmc_modes[ ,2:ncol(mcmc_modes)])

cov_modes[upper.tri(cov_modes)] <- cor_modes[upper.tri(cor_modes)]

pandoc.table(cov_modes)
```

	mcmc.mode.1	mcmc.mode.2	mcmc.mode.3
mcmc.mode.1	12.03	0.9508	0.9484
mcmc.mode.2	11.5	12.16	0.9524

	mcmc.mode.1	mcmc.mode.2	mcmc.mode.3
mcmc.mode.3	11.55	11.66	12.33
mcmc.mode.4	11.57	11.65	11.71
mcmc.mode.5	11.55	11.62	11.65
mcmc.mode.6	11.51	11.63	11.61
mcmc.mode.7	11.58	11.72	11.72
mcmc.mode.8	11.47	11.56	11.52

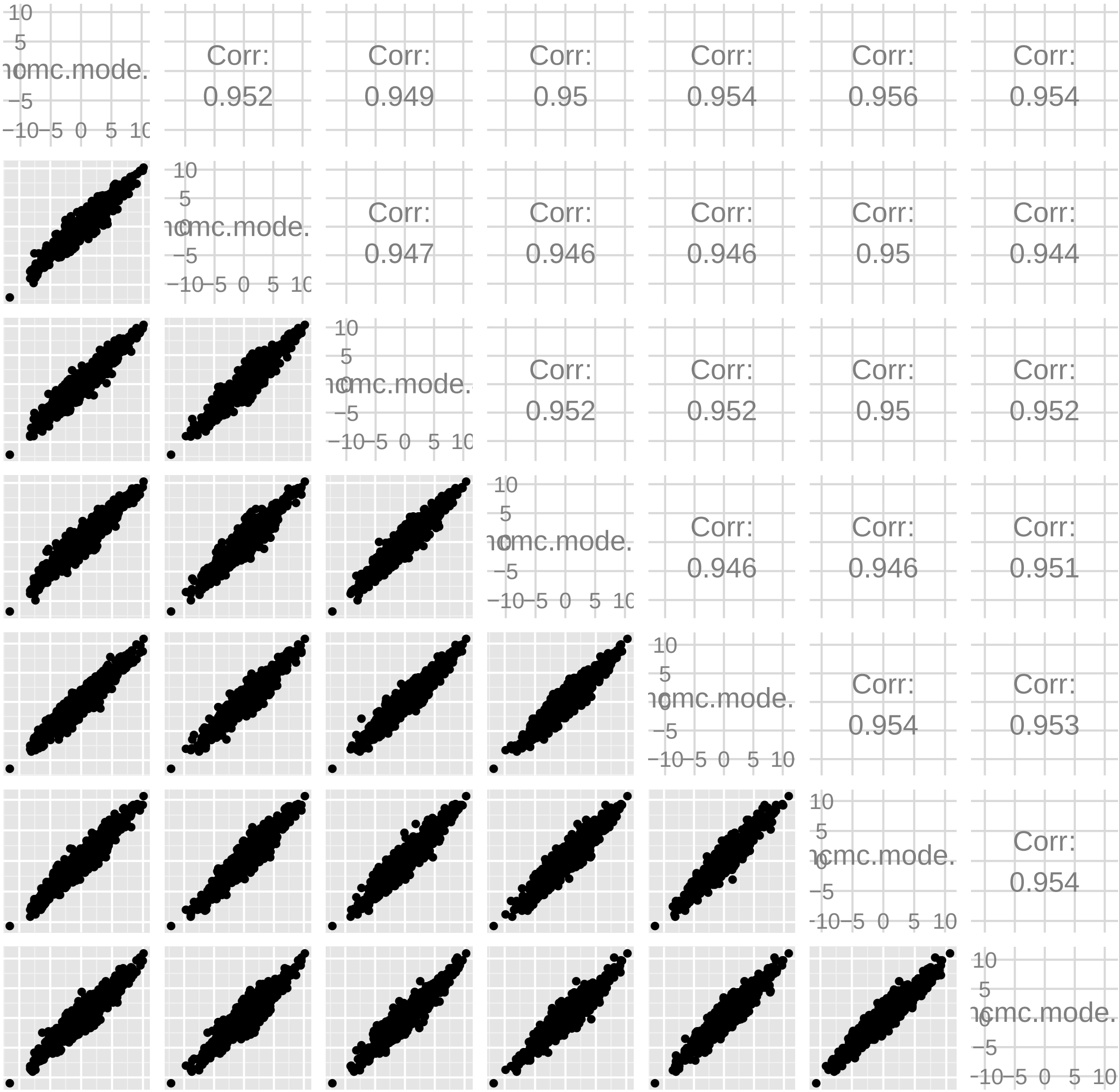
Table 14: Table continues below

	mcmc.mode.4	mcmc.mode.5	mcmc.mode.6
mcmc.mode.1	0.9475	0.9488	0.9495
mcmc.mode.2	0.9488	0.9498	0.9542
mcmc.mode.3	0.9473	0.9458	0.9464
mcmc.mode.4	12.4	0.9516	0.9522
mcmc.mode.5	11.76	12.31	0.9463
mcmc.mode.6	11.72	11.6	12.21
mcmc.mode.7	11.76	11.67	11.72
mcmc.mode.8	11.66	11.6	11.58

Table 15: Table continues below

	mcmc.mode.7	mcmc.mode.8
mcmc.mode.1	0.9496	0.9516
mcmc.mode.2	0.9562	0.9536
mcmc.mode.3	0.9497	0.9439
mcmc.mode.4	0.95	0.9524
mcmc.mode.5	0.9463	0.9511
mcmc.mode.6	0.9541	0.953
mcmc.mode.7	12.36	0.9543
mcmc.mode.8	11.66	12.08

```
library(ggplot2)
library(GGally)
ggpairs(mcmc_modes, columns = 3:ncol(mcmc_modes))
```


Corr:
0.952
Corr:
0.949
Corr:
0.95
Corr:
0.954
Corr:
0.956
Corr:
0.954
Corr:
0.947
Corr:
0.946
Corr:
0.946
Corr:
0.95
Corr:
0.944
Corr:
0.952
Corr:
0.952
Corr:
0.95
Corr:
0.952
Corr:
0.946
Corr:
0.946
Corr:
0.951
Corr:
0.954
Corr:
0.953
Corr:
0.954
mcmc.mode.
10
5
0
−5
−10 −5 0 5 10

Ok, the models are all converging on the same point estimates. Why 0.95 correlation???

Variance of blups

```
mcmc_itts <- mcmc_priors[!is.na(mcmc_priors$itt), ]
tapply(mcmc_itts$docility, mcmc_itts$type, var)
```

```
## mcmc.1 mcmc.2 mcmc.3 mcmc.4 mcmc.5 mcmc.6 mcmc.7 mcmc.8
##  19.20  19.30  19.16  19.12  19.20  19.13  19.26  19.14
```

```
tapply(mcmc_itts$docility, mcmc_itts$type, range)
```

```
## $mcmc.1
## [1] -18.35  18.04
##
## $mcmc.2
## [1] -20.60  19.28
##
## $mcmc.3
## [1] -20.06  18.31
##
## $mcmc.4
## [1] -20.17  18.76
##
## $mcmc.5
## [1] -19.19  17.83
##
## $mcmc.6
## [1] -19.51  18.44
##
## $mcmc.7
## [1] -18.99  18.26
##
## $mcmc.8
## [1] -22.01  17.66
```

No variation in variances either…

Fitness and Standardization

Load Data

```
library(MASS) # MASS clashes with dplyr... so always load first
library(pander) # pander clashes with dplyr... so always load first
```

```
##
## Attaching package: 'pander'
##
```

```
## The following object is masked from 'package:knitr':
##
##     pandoc
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:MASS':
##
##     select
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.alignment('right', row.names = 'left')
load("data/analyses_data/raneffs_blups.RData")
# load("data/analyses_data/pca.RData")
fitness <- read.table(file = "data/fitness+competition.csv", sep = ',',
  header = TRUE, stringsAsFactors = FALSE)
fitness <- tbl_df(fitness)
```

Merge fitness with blups

```
fitness$ID <- as.character(fitness$ID)
fit_raneff_data <- inner_join(fitness, raneffs_blups, by = "ID")
fit_raneff_data <- fit_raneff_data %>%
  filter(!is.na(docility) & !is.na(aggression) & !is.na(activity))
```

Relative Fitness

Calculate relative fitness for each year & population combination. Two populations (Grids). Three measures of fitness:

1. ars_all = Annual reproductive success over all litters (no. pups that survived overwinter)

2. kprod = Fecundity (kids produced)

3. prop = Offpsring overwinter survival (proportion of pups produced that survived overwinter)

```
# Calculate offspring overwitner survival
fit_raneff_data <- fit_raneff_data %>% mutate(prop = ars_all/kprod)
```

```
fit_raneff_data <- fit_raneff_data %>%
  group_by(Grid, Year, itt) %>%
  mutate(rel_ars = ars_all / mean(ars_all),
   rel_kpd = kprod / mean(kprod),
   rel_ows = prop / mean(prop)
   )
```

Now:

- rel_ars = relative ARS
- rel_kpd = relative fecunidty
- felOWS = relative offspring overwinter survival

Standardize Variables

Standaradized to mean 0 and sd 1. Standardized variables renamed from xxx to xxx_s or xxx_sy (for standardized within year). Standardized within each BLUP itteration.

```
# Standardize within itteration and year
fit_raneff_data <- fit_raneff_data %>%
  group_by(itt, Year, add = FALSE) %>%
  mutate(
    aggression_sy  = (aggression  - mean(aggression, na.rm = TRUE))  /
      sd(aggression, na.rm = TRUE),
    activity_sy    = (activity    - mean(activity, na.rm = TRUE))    /
      sd(activity, na.rm = TRUE),
    docility_sy    = (docility    - mean(docility, na.rm = TRUE))    /
      sd(docility, na.rm = TRUE),
    competition_sy = (competition - mean(competition, na.rm = TRUE)) /
      sd(competition, na.rm = TRUE)
      )
# Standardize within itteration
fit_raneff_data <- fit_raneff_data %>%
  group_by(itt, add = FALSE) %>%
  mutate(
    aggression_s  = (aggression  - mean(aggression, na.rm = TRUE))  /
      sd(aggression, na.rm = TRUE),
    activity_s    = (activity    - mean(activity, na.rm = TRUE))    /
      sd(activity, na.rm = TRUE),
    docility_s    = (docility    - mean(docility, na.rm = TRUE))    /
      sd(docility, na.rm = TRUE),
    competition_s = (competition - mean(competition, na.rm = TRUE)) /
      sd(competition, na.rm = TRUE)
      )

fit_raneff_data %>%
  group_by(itt, add = FALSE) %>%
  summarise(
    v_agg = var(aggression_s, na.rm = TRUE),
```

```
    v_act = var(activity_s, na.rm = TRUE),
    v_doc = var(docility_s, na.rm = TRUE)
    ) %>%
  head(.,n=10) %>%
  pandoc.table(.)
```

itt	v_agg	v_act	v_doc
0	1	1	1
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1
8	1	1	1
9	1	1	1

```
fit_raneff_data %>%
  group_by(itt, Year, add = FALSE) %>%
  summarise(
    v_agg = var(aggression_sy, na.rm = TRUE),
    v_act = var(activity_sy, na.rm = TRUE),
    v_doc = var(docility_sy, na.rm = TRUE)
    ) %>%
  head(.,n=10) %>%
  pandoc.table(.)
```

itt	Year	v_agg	v_act	v_doc
0	2003	1	1	1
0	2004	1	1	1
0	2005	1	1	1
0	2006	1	1	1
0	2007	1	1	1
0	2008	1	1	1
0	2009	1	1	1
0	2010	1	1	1
1	2003	1	1	1
1	2004	1	1	1

Sample Sizes

```
fit_raneff_data %>%
  filter(itt == "1") %>%
  group_by(Grid, Year, add = FALSE) %>%
  summarise(n()) %>%
  pandoc.table(.)
```

Grid	Year	n()
KL	2003	3
KL	2004	6
KL	2005	15
KL	2006	19
KL	2007	20
KL	2008	28
KL	2009	24
KL	2010	16
SU	2003	10
SU	2004	15
SU	2005	26
SU	2006	21
SU	2007	14
SU	2008	9
SU	2009	9
SU	2010	2

```
fit_raneff_data %>%
filter(itt == "1") %>%
  group_by(Year, add = FALSE) %>%
  summarise(n()) %>%
  pandoc.table(.)
```

Year	n()
2003	13
2004	21
2005	41
2006	40
2007	34
2008	37

Year	n()
2009	33
2010	18

```
save(fit_raneff_data, file = "data/analyses_data/fit_raneff_data.RData")
```

Temporal Selection Gradients

```
library(MASS) # MASS clashes with dplyr... so always load first
library(pander) # pander clashes with dplyr... so always load first
```

```
##
## Attaching package: 'pander'
##
## The following object is masked from 'package:knitr':
##
##     pandoc
```

```
set.alignment('right', row.names = 'left')
library(MCMCglmm)
```

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

```
library(arm)
```

```
## Loading required package: lme4
## Loading required package: Rcpp
##
## arm (Version 1.7-07, built: 2014-8-27)
##
## Working directory is /home/ryan/projects/2014-female-selection
##
##
## Attaching package: 'arm'
##
## The following object is masked from 'package:ape':
##
##     balance
##
## The following object is masked from 'package:coda':
##
##     traceplot
```

```
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:MASS':
##
##     select
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(grid)

# Load Data
load("data/analyses_data/fit_raneff_data.RData")
fit_raneff_data$Year <- as.character(fit_raneff_data$Year)
```

Does 'Year' significantly improve models of selection?

The significance of interaction terms between Year and the behavioral traits on fitness.

```
# GLMMs to test whether selection fluctuates across years.
library(lme4)
# Models with interactions between year and the behavioral traits
fit_raneff_data_blup <- filter(fit_raneff_data, type == "blup")

fit_raneff_data_blup %>%
  group_by(Grid, Year, add = FALSE) %>%
  summarise(n(), var(ars_all))

## Source: local data frame [16 x 4]
## Groups: Grid
##
##    Grid Year n() var(ars_all)
## 1    KL 2003   4       0.0000
## 2    KL 2004   8       0.7857
## 3    KL 2005  19       1.6550
## 4    KL 2006  24       0.4275
## 5    KL 2007  21       1.1000
## 6    KL 2008  29       0.5369
## 7    KL 2009  24       0.7373
## 8    KL 2010  22       1.2121
## 9    SU 2003  14       1.3022
```

```
## 10   SU 2004  18       0.8007
## 11   SU 2005  31       1.9828
## 12   SU 2006  24       0.3025
## 13   SU 2007  19       1.0526
## 14   SU 2008  16       0.0000
## 15   SU 2009  11       0.8727
## 16   SU 2010  12       2.4470
```

```
# grid_years with no variation in fitness need to be removed
fit_raneff_data_blup <- filter(fit_raneff_data_blup,
  !(grid_year %in% c("KL2003", "SU2008")))

# Need to also remove Grid Years with very low sample sizes.
fit_raneff_data_blup %>%
  group_by(Grid, Year, add = FALSE) %>%
  summarise(n(), var(ars_all))
```

```
## Source: local data frame [14 x 4]
## Groups: Grid
##
##    Grid Year n() var(ars_all)
## 1    KL 2004   8       0.7857
## 2    KL 2005  19       1.6550
## 3    KL 2006  24       0.4275
## 4    KL 2007  21       1.1000
## 5    KL 2008  29       0.5369
## 6    KL 2009  24       0.7373
## 7    KL 2010  22       1.2121
## 8    SU 2003  14       1.3022
## 9    SU 2004  18       0.8007
## 10   SU 2005  31       1.9828
## 11   SU 2006  24       0.3025
## 12   SU 2007  19       1.0526
## 13   SU 2009  11       0.8727
## 14   SU 2010  12       2.4470
```

```
fit_raneff_data_blup <- filter(fit_raneff_data_blup,
  !(grid_year %in% c("KL2004")))

fit_raneff_data_blup$oID <- 1:nrow(fit_raneff_data_blup)
fit_raneff_data_blup <- droplevels(fit_raneff_data_blup)

ars_year <- glmer(ars_all ~ Year + Grid + activity_s + activity_s:Year +
  aggression_s + aggression_s:Year + docility_s + docility_s:Year +
  (1|ID) + (1|oID), data = fit_raneff_data_blup, family = poisson,
  control=glmerControl(optimizer="bobyqa"))
```

```
## Warning: maxfun < 10 * length(par)^2 is not recommended.
```

```
save(ars_year, file = "data/analyses_data/ars_year.RData")
```

```r
load("data/analyses_data/ars_year.RData")
# Test fit of models. Does the addition of year:traits improve the fit?
library(car)
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:arm':
##
##     logit
```

```r
library(lme4)
aod <- Anova(ars_year, type = 2)
aod <- data.frame(aod)
aod$Chisq <- round(aod$Chisq ,digits = 2)

# Function to convert small p-values into 'P < X'
p.format <- function(x){
    out <- signif(x, digits = 2)
  out[x < 0.005]  <- "< 0.005"
    out[x < 0.001]  <- "< 0.001"
    out[x < 0.0001] <- "< 0.0001"
    return(out)
}
# Format p values
aod[ ,3] <- p.format(aod[ ,3])
row.names(aod) <- c("Year", "Grid", "Activity", "Aggression", "Docility",
  "Year x Activity", "Year x Aggression", "Year x Docility")
pandoc.table(aod, caption =
  "The effect of year on selection for female behavioral traits through annual
reproductive success. Significance was calculated with Wald chisq tests from
a type II analysis of deviance. GLMMs were fitted with identity as a random
effect and assumed a Poisson error distribution.")
```

	Chisq	Df	Pr..Chisq.
Year	38.1	7	< 0.0001
Grid	1.07	1	0.3
Activity	0.34	1	0.56
Aggression	0.34	1	0.56
Docility	0.98	1	0.32
Year x Activity	22.92	7	< 0.005
Year x Aggression	22.84	7	< 0.005
Year x Docility	10.33	7	0.17

	Chisq	Df	Pr..Chisq.

Table 21: The effect of year on selection for female behavioral traits through annual reproductive success. Significance was calculated with Wald chisq tests from a type II analysis of deviance. GLMMs were fitted with identity as a random effect and assumed a Poisson error distribution.

Accounting for random effect uncertainty

Calculate selection gradients

Calculate selection coefficients for each of the 1000 samples of the posterior distribution of random effects.

```
# Function to get posterior distribution of selection gradients
x <- fit_raneff_data %>% filter(type == "raneff" & Year == 2003 & itt == 1)
seCoefMCMC <- function(x){
  mod <- lm(rel_ars ~ aggression + activity + docility, data = x)
  mod_sd <-  lm(rel_ars ~ aggression_sy + activity_sy + docility_sy, data = x)
  res <- c(as.list(coef(mod)[-1]), as.list(coef(mod_sd)[-1]))
  res$Year <- x$Year[1]
  res$itt <- x$itt[1]
  return(data.frame(res, stringsAsFactors = FALSE))
}

start_time <- Sys.time()
sel_grads_mcmc_post <- fit_raneff_data %>%
  filter(type == "raneff") %>%
  group_by(itt, Year, add = FALSE) %>%
  do(seCoefMCMC(.))
print(paste("Approx. run time: ", format(Sys.time() - start_time)))

## [1] "Approx. run time:  57.5 secs"

save(sel_grads_mcmc_post,
  file = "data/analyses_data/sel_grads_mcmc_post.RData")

load("data/analyses_data/sel_grads_mcmc_post.RData")
x <- sel_grads_mcmc_post %>% filter(Year == "2003")
getCred <- function(x, sig = 0.05){
  require(MCMCglmm)
  mcmc_data <- x %>% ungroup() %>% select(aggression, activity, docility, aggression_sy, activity_sy
  pm  <- posterior.mode(mcmc_data)
  int <- HPDinterval(mcmc_data, prob = 1 - sig)
  tbl_df(data.frame(
    Year      = x$Year[1],
    variable  = c("Aggression", "Activity", "Docility", "Aggression",
    "Activity", "Docility"),
```

```
     standardization = c("None", "None", "None", "SD", "SD", "SD"),
    post_mode = pm,
    lower     = int[ ,"lower"],
    upper     = int[ ,"upper"],
    stringsAsFactors = FALSE
    ))
}

getCred(sel_grads_mcmc_post %>% filter(Year == "2004"))

## Source: local data frame [6 x 6]
##
##               Year   variable standardization post_mode    lower   upper
## aggression    2004 Aggression            None  -0.30391 -0.76486 0.14086
## activity      2004   Activity            None   0.03346 -0.31909 0.66905
## docility      2004   Docility            None   0.05488 -0.06433 0.11222
## aggression_sy 2004 Aggression              SD  -0.24452 -0.80615 0.07615
## activity_sy   2004   Activity              SD   0.01654 -0.35851 0.62341
## docility_sy   2004   Docility              SD   0.20932 -0.25839 0.51523

sel_grads_mcmc <- sel_grads_mcmc_post %>%
  group_by(Year, add = FALSE) %>%
  do(getCred(x = ., sig = 0.05))

sel_grads_mcmc$upper_sig_star <- ""
sel_grads_mcmc$lower_sig_star <- ""
sel_grads_mcmc$upper_sig_star[sel_grads_mcmc$post_mode > 0 &
  sel_grads_mcmc$lower > 0] <- "*"
sel_grads_mcmc$lower_sig_star[sel_grads_mcmc$post_mode < 0 &
  sel_grads_mcmc$upper < 0] <- "*"

save(sel_grads_mcmc, sel_grads_mcmc_post, getCred,
  file = "data/analyses_data/sel_grads_mcmc.RData")

load("data/analyses_data/sel_grads_mcmc.RData")

N <- fit_raneff_data %>%
  filter(type == "blup") %>%
  group_by(Year, add = FALSE) %>%
  summarise(n(), doc_mean = mean(docility, na.rm = TRUE))

# Format for table
sgt <- sel_grads_mcmc
sgt$sig_star <- ""
sgt$sig_star[sgt$post_mode > 0 & sgt$lower > 0] <- "*"
sgt$sig_star[sgt$post_mode < 0 & sgt$upper < 0] <- "*"
sgt$post_mode <- format(round(sgt$post_mode, digits = 2), digits = 1,
  nsmall = 2)
sgt$lower     <- format(round(sgt$lower,     digits = 2), digits = 1,
  nsmall = 2)
sgt$upper     <- format(round(sgt$upper,     digits = 2), digits = 1,
```

```
  nsmall = 2)
sgt$coef      <- paste(sgt$post_mode, " (", sgt$lower, " to ", sgt$upper,")",
  sgt$sig_star, sep = '')

sgt_agg <- filter(sgt, variable == "Aggression", standardization == "None")
sgt_act <- filter(sgt, variable == "Activity",   standardization == "None")
sgt_doc <- filter(sgt, variable == "Docility",   standardization == "None")

sgt_agg_sd <- filter(sgt, variable == "Aggression", standardization == "SD")
sgt_act_sd <- filter(sgt, variable == "Activity",   standardization == "SD")
sgt_doc_sd <- filter(sgt, variable == "Docility",   standardization == "SD")


doc_post_mode <- sel_grads_mcmc %>% filter(standardization == "None" & variable == "Docility")
doc_post_mode$post_mode_m <- doc_post_mode$post_mode * N$doc_mean
doc_post_mode$post_mode_m <- format(round(doc_post_mode$post_mode_m, digits = 2), digits = 1, nsmall

pandoc.table(
  data.frame(Year = N$Year, N = N[ ,2], Aggression = sgt_agg$coef,
    Acitivity = sgt_act$coef, Docility = sgt_doc$coef
  ),
  caption = "Non-standardized selection gradients (accounting for behavioural uncertainty)."
  )
```

Year	N	Aggression	Acitivity
2003	18	0.03 (-0.92 to 0.84)	0.45 (-0.17 to 1.32)
2004	26	-0.30 (-0.76 to 0.14)	0.03 (-0.32 to 0.67)
2005	50	-0.18 (-0.45 to 0.10)	0.13 (-0.16 to 0.40)
2006	48	0.39 (-0.04 to 0.79)	0.08 (-0.39 to 0.45)
2007	40	0.00 (-0.35 to 0.36)	0.11 (-0.22 to 0.42)
2008	45	0.44 (-0.12 to 0.95)	-0.26 (-0.75 to 0.40)
2009	35	0.12 (-0.33 to 0.58)	-0.58 (-0.93 to -0.02)*
2010	34	0.06 (-0.30 to 0.35)	0.03 (-0.26 to 0.30)

Table 22: Non-standardized selection gradients (accounting for behavioural uncertainty). (continued below)

Docility
-0.08 (-0.26 to 0.06)
0.05 (-0.06 to 0.11)
0.03 (-0.02 to 0.07)
-0.04 (-0.12 to 0.00)
0.00 (-0.05 to 0.05)

Docility
-0.02 (-0.08 to 0.05)
-0.06 (-0.11 to -0.01)*
-0.05 (-0.08 to -0.02)*

```
pandoc.table(
  data.frame(Year = N$Year, N = N[ ,2], Aggression = sgt_agg_sd$coef,
    Acitivity = sgt_act_sd$coef, Docility = sgt_doc_sd$coef
  ),
  caption = "SD-standardized selection gradients (accounting for behavioural uncertainty)."
  )
```

Year	N	Aggression	Acitivity
2003	18	0.03 (-0.76 to 0.71)	0.79 (-0.24 to 1.38)
2004	26	-0.24 (-0.81 to 0.08)	0.02 (-0.36 to 0.62)
2005	50	-0.19 (-0.47 to 0.07)	0.16 (-0.12 to 0.41)
2006	48	0.39 (-0.04 to 0.72)	0.08 (-0.37 to 0.44)
2007	40	-0.01 (-0.32 to 0.34)	0.09 (-0.23 to 0.42)
2008	45	0.46 (-0.06 to 0.95)	-0.18 (-0.72 to 0.40)
2009	35	0.16 (-0.29 to 0.59)	-0.55 (-0.92 to 0.01)
2010	34	-0.04 (-0.27 to 0.32)	0.04 (-0.25 to 0.30)

Table 24: SD-standardized selection gradients (accounting for behavioural uncertainty). (continued below)

Docility
-0.38 (-1.12 to 0.31)
0.21 (-0.26 to 0.52)
0.10 (-0.09 to 0.34)
-0.25 (-0.58 to -0.03)*
0.01 (-0.23 to 0.23)
-0.08 (-0.37 to 0.25)
-0.30 (-0.53 to -0.05)*
-0.23 (-0.35 to -0.10)*

```
pandoc.table(
  data.frame(Year = N$Year, N = N[ ,2], mean_trait = N$doc_mean, Docility = doc_post_mode$post_mode_
```

```
  ),
  caption = "Mean standardized selection gradients (accounting for behavioural uncertainty)."
  )
```

Year	N	mean_trait	Docility
2003	18	17.29	-1.46
2004	26	17.31	0.95
2005	50	17.25	0.55
2006	48	16.77	-0.70
2007	40	16.77	0.05
2008	45	16.9	-0.27
2009	35	16.79	-0.97
2010	34	17.19	-0.87

Table 26: Mean standardized selection gradients (accounting for behavioural uncertainty).

Plots

```
load("data/analyses_data/sel_grads_mcmc_post.RData")
library(ggplot2)
library(dplyr)

sel_grads_mcmc <- sel_grads_mcmc_post %>%
  group_by(Year, add = FALSE) %>% do(getCred(x = ., sig = 0.05))

sel_grads_mcmc$upper_sig_star <- ""
sel_grads_mcmc$lower_sig_star <- ""
sel_grads_mcmc$upper_sig_star[sel_grads_mcmc$post_mode > 0 &
  sel_grads_mcmc$lower > 0] <- "*"
sel_grads_mcmc$lower_sig_star[sel_grads_mcmc$post_mode < 0 &
  sel_grads_mcmc$upper < 0] <- "*"

p <- ggplot(data = filter(sel_grads_mcmc, standardization == "SD"),
  aes(x = Year, y = post_mode, group = variable))
p <- p + geom_hline(yintercept = 0, size = 0.25) # Line at y = 0
p <- p + geom_errorbar(aes(ymax = upper, ymin = lower),
  position = position_dodge(width = 0.5), width = 0.4, size = 0.4)
# Houle data percentiles
p <- p + geom_hline(yintercept = c(0.2975, -0.2975), linetype = 2, size = 0.4)
p <- p + geom_point(aes(shape = variable, fill = variable),
  position = position_dodge(width = 0.5), size = 3)
p <- p + scale_shape_manual(name = "B", values = c(24, 21, 22))
p <- p + scale_fill_manual(name = "B", values = c("white", "black", "white"))
p <- p + scale_color_manual(name = "B", values = c("black", "black", "black"))
```

```r
p <- p + xlab("Year")
p <- p + ylab("Posterior Mode ± 0.95 Credible Interval")
p <- p + theme_bw(base_size = 10)
p <- p + theme(legend.position = c(0.92, 0.86),
  legend.background = element_blank(), legend.key.size = unit(0.4, "cm"))
p <- p + theme(legend.title = element_text(family = "Helvetica",
                                    face = "plain", size = 18))
p <- p + theme(legend.key = element_blank())
p <- p + theme(strip.background = element_blank())
p <- p + theme(panel.grid.minor = element_blank(),
  panel.grid.major = element_blank())
p <- p + theme(panel.border = element_blank())
p <- p + theme(axis.line = element_line(color = "black"))
p <- p + geom_text(aes(x = Year, y = upper, group = variable,
  label = upper_sig_star), vjust = -0.3,
  position = position_dodge(width = 0.5), size = 5)
p <- p + geom_text(aes(x = Year, y = lower, group = variable,
  label = lower_sig_star), vjust = 1.3,
  position = position_dodge(width = 0.5), size = 5)
p_sel_grad_MCMC <- p + ylim(c(-1.1,1.4))

pdf(file = "figure/04_sg_mcmc_SD_print.pdf", width = 4.33, height = 3)
p_sel_grad_MCMC

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 1 rows containing missing values (geom_text).

dev.off()

## pdf
##   2

p_sel_grad_MCMC

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 1 rows containing missing values (geom_text).


load("data/analyses_data/sel_grads_mcmc_post.RData")
library(ggplot2)
library(dplyr)
```

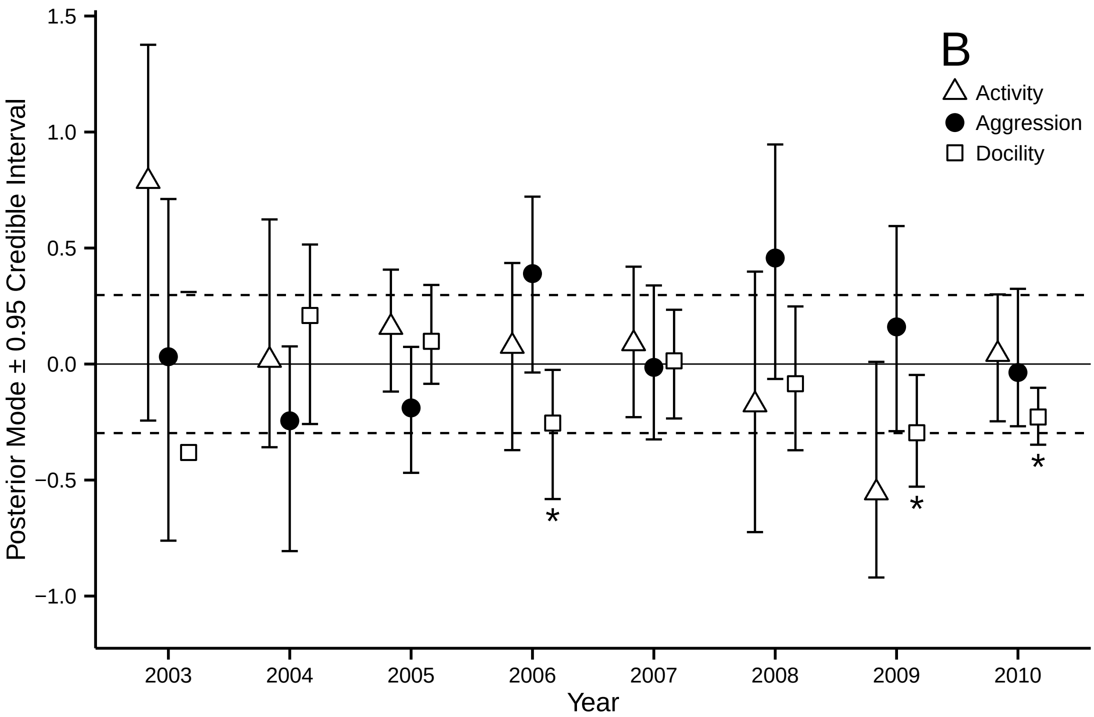


Figure 1: SD Standardized Selection Gradients

```
sel_grads_mcmc <- sel_grads_mcmc_post %>%
  group_by(Year, add = FALSE) %>% do(getCred(x = ., sig = 0.05))

sel_grads_mcmc$upper_sig_star <- ""
sel_grads_mcmc$lower_sig_star <- ""
sel_grads_mcmc$upper_sig_star[sel_grads_mcmc$post_mode > 0 &
  sel_grads_mcmc$lower > 0] <- "*"
sel_grads_mcmc$lower_sig_star[sel_grads_mcmc$post_mode < 0 &
  sel_grads_mcmc$upper < 0] <- "*"

p <- ggplot(data = filter(sel_grads_mcmc, standardization == "None"),
  aes(x = Year, y = post_mode, group = variable)
)
p <- p + geom_hline(yintercept = 0, size = 0.25) # Line at y = 0
p <- p + geom_errorbar(aes(ymax = upper, ymin = lower),
  position = position_dodge(width = 0.5), width = 0.4, size = 0.4)
# Houle data percentiles
p <- p + geom_hline(yintercept = c(0.2975, -0.2975), linetype = 2, size = 0.4)
p <- p + geom_point(aes(shape = variable, fill = variable),
  position = position_dodge(width = 0.5), size = 3)
p <- p + scale_shape_manual(name = "B", values = c(24, 21, 22))
p <- p + scale_fill_manual(name = "B", values = c("white", "black", "white"))
p <- p + scale_color_manual(name = "B", values = c("black", "black", "black"))
p <- p + xlab("Year")
p <- p + ylab("Posterior Mode ± 0.95 Credible Interval")
p <- p + theme_bw(base_size = 10)
p <- p + theme(legend.position = c(0.92, 0.86),
  legend.background = element_blank(), legend.key.size = unit(0.4, "cm"))
p <- p + theme(legend.title = element_text(family = "Helvetica",
                                face = "plain", size = 18))
p <- p + theme(legend.key = element_blank())
p <- p + theme(strip.background = element_blank())
p <- p + theme(panel.grid.minor = element_blank(),
  panel.grid.major = element_blank())
p <- p + theme(panel.border = element_blank())
p <- p + theme(axis.line = element_line(color = "black"))
p <- p + geom_text(aes(x = Year, y = upper, group = variable,
  label = upper_sig_star), vjust = -0.3,
  position = position_dodge(width = 0.5), size = 5)
p <- p + geom_text(aes(x = Year, y = lower, group = variable,
  label = lower_sig_star), vjust = 1.3,
  position = position_dodge(width = 0.5), size = 5)
p_sel_grad_MCMC <- p + ylim(c(-1.1,1.4))

pdf(file = "figure/04_sg_mcmc_NS_print.pdf", width = 4.33, height = 3)
p_sel_grad_MCMC

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

dev.off()
```

```
## pdf
##   2
```

```
p_sel_grad_MCMC
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```

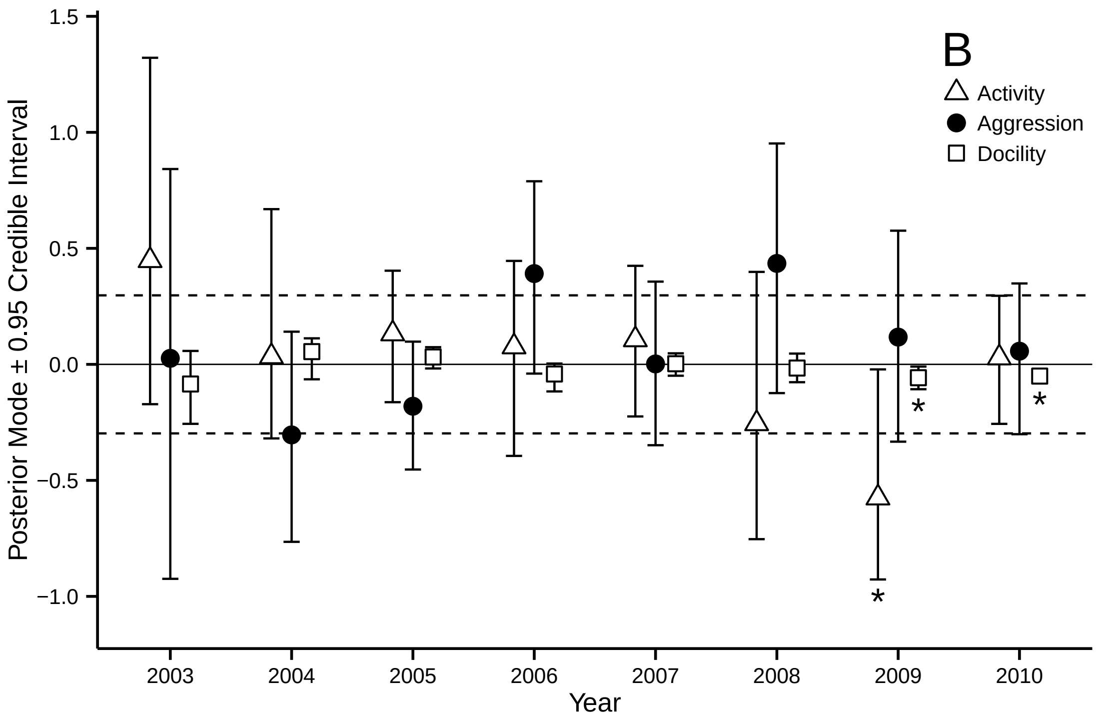


Figure 2: Non-standardized Selection Gradients

Correlations

```
sel_grads_mcmc_flat <- data.frame(
  Aggression = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Aggression")$post_mode,
  Agg_upper  = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Aggression")$upper,
  Agg_lower  = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Aggression")$lower,
  Activity   = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Activity")$post_mode,
  Act_upper  = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Activity")$upper,
```

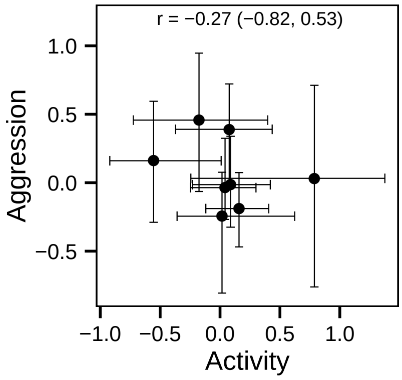
```
  Act_lower  = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Activity")$lower,
  Docility   = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Docility")$post_mode,
  Doc_upper  = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Docility")$upper,
  Doc_lower  = filter(sel_grads_mcmc, standardization == "SD",
   variable == "Docility")$lower
)

cor.behav <- function(x, y){
  ct <- cor.test(x, y)
  out <- data.frame(est = ct$estimate, lower = ct$conf.int[1],
    upper = ct$conf.int[2], stringsAsFactors = FALSE)
  out <- round(out, digits = 2)
  out$print <- paste(out$est, " (", out$lower, ", ", out$upper, ")", sep = "")
}

cor_agg_act <- cor.behav(sel_grads_mcmc_flat$Aggression,
  sel_grads_mcmc_flat$Activity)
cor_agg_doc <- cor.behav(sel_grads_mcmc_flat$Aggression,
  sel_grads_mcmc_flat$Docility)
cor_doc_act <- cor.behav(sel_grads_mcmc_flat$Docility,
  sel_grads_mcmc_flat$Activity)
```

Aggression and Activity

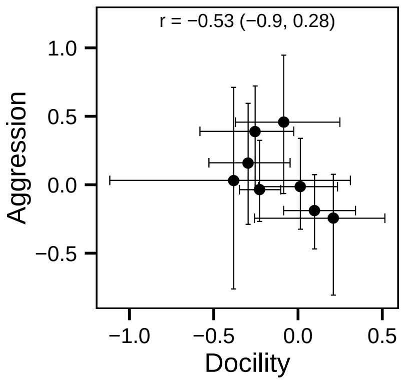
```
p <- ggplot(data = sel_grads_mcmc_flat, aes(x = Activity, y = Aggression))
p <- p + geom_point()
p <- p + ylab("Aggression")
p <- p + xlab("Activity")
p <- p + theme_bw(base_size = 10)
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + geom_errorbarh(aes(xmin = Act_lower, xmax = Act_upper),
  height = 0.07, size = 0.2)
p <- p + geom_errorbar(aes(ymin = Agg_lower, ymax = Agg_upper),
  width = 0.07, size = 0.2)
p <- p + annotate(geom = "text", size = 2.5, x = 0.25, y = 1.2,
  label = paste("r = ", cor_agg_act, sep = ''))
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```

Aggression and Docility

```
p <- ggplot(data = sel_grads_mcmc_flat, aes(y = Aggression, x = Docility))
p <- p + geom_point()
p <- p + ylab("Aggression")
```



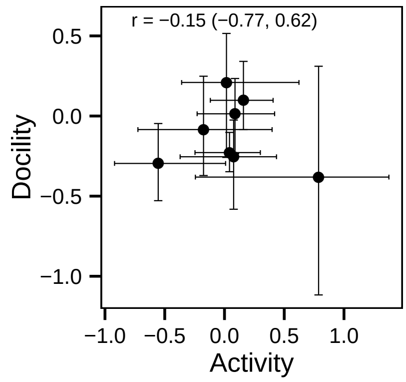

```
p <- p + xlab("Docility")
p <- p + theme_bw(base_size = 10)
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + geom_errorbarh(aes(xmin = Doc_lower, xmax = Doc_upper),
  height = 0.07, size = 0.2)
p <- p + geom_errorbar(aes(ymin = Agg_lower, ymax = Agg_upper),
  width = 0.03, size = 0.2)
p <- p + annotate(geom = "text", size = 2.5, x = -0.3, y = 1.2,
  label = paste("r = ", cor_agg_doc, sep = ''))
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Activity and Docility

```
p <- ggplot(data = sel_grads_mcmc_flat, aes(x = Activity, y = Docility))
p <- p + geom_point()
p <- p + xlab("Activity")
p <- p + ylab("Docility")
p <- p + theme_bw(base_size = 10)
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
```

```
  strip.background = element_blank(), strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + geom_errorbar(aes(ymin = Doc_lower, ymax = Doc_upper),
  width = 0.07, size = 0.2)
p <- p + geom_errorbarh(aes(xmin = Act_lower, xmax = Act_upper),
  height = 0.03, size = 0.2)
p <- p + annotate(geom = "text", size = 2.5, x = 0, y = 0.6,
  label = paste("r = ", cor_doc_act, sep = ''))
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Summary Stats

```
library(dplyr)
sign_change <- function(x){
  # sign changes calculated as the number of changes in direction between
  # successive years relative to n-1
  s <- sign(x)
  sum(s[1:(length(s)-1)] != s[-1])/(length(s)-1)
}
se <- function(lower, upper){
  (upper - lower) / 3.92
}

sum_stats_mcmc <- sel_grads_mcmc %>%
filter(standardization == "SD") %>%
group_by(variable, add = FALSE) %>%
summarise(
  mean_abs_b = mean(abs(post_mode)),
  abs_mean_b = abs(mean(post_mode)),
  sd_b = sd(post_mode),
  mean_se_b = mean(se(lower, upper)),
  freq_sign = sign_change(post_mode),
  mean_cv = mean(se(lower, upper) / abs(post_mode))
  )
sum_stats_mcmc[ ,2:6] <- round(sum_stats_mcmc[ ,2:6], 2)
pandoc.table(sum_stats_mcmc)
```

variable	mean_abs_b	abs_mean_b	sd_b	mean_se_b
Activity	0.24	0.05	0.37	0.23
Aggression	0.19	0.07	0.25	0.22
Docility	0.2	0.12	0.21	0.16

Table 27: Table continues below

freq_sign	mean_cv
0.29	3.307
0.71	4.002
0.57	1.845

Ignoring random effect uncertainty

Calculate selection gradients from BLUPs, estimate SE using jackknifing.

Calculate selection gradients

```
# Calculate standardized selection gradients
seCoeffLmer <- function(x){
  model <- lm(rel_ars ~ aggression + activity + docility, data = x)
  model_sy <- lm(rel_ars ~ aggression_sy + activity_sy + docility_sy, data = x)
  mod_coefs <- c(coef(model)[-1], coef(model_sy)[-1])
  sim_coefs <- data.frame(coef(sim(model))[ ,-1], coef(sim(model_sy))[ ,-1]) ## Simulated coefficien
  names(sim_coefs) <- names(mod_coefs)
  sim_CI <- apply(sim_coefs, 2, quantile, prob = c(0.025, 0.975)) #0.95 conf. int.

docil_mean_coef <- mod_coefs["docility"] * mean(x$docility, na.rm = TRUE)
  out <- data.frame(
    standardization = c("None","None","None","SD","SD","SD","Mean"),
    Year = as.numeric(rep(as.character(x$Year[1]), 7)),
    variable = c("Aggression", "Activity", "Docility", "Aggression",
      "Activity", "Docility", "Docility"),
    coefficients = c(mod_coefs, docil_mean_coef),
    lower = c(sim_CI[1, ], 0),
    upper = c(sim_CI[2, ],0)
    )
    return(out)
}

sel_grads_blup <- fit_raneff_data %>%
  filter(type == "blup") %>%
  group_by(Year, add = FALSE) %>%
  do(seCoeffLmer(.))
```

```
sel_grads_blup$variable <- as.character(sel_grads_blup$variable)
sel_grads_blup$Year <- as.character(sel_grads_blup$Year)
sel_grads_blup <- tbl_df(sel_grads_blup)
save(sel_grads_blup, file = "data/analyses_data/sel_grads_blup.RData")

load("data/analyses_data/sel_grads_blup.RData")
# Format for table
sgt <- sel_grads_blup
sgt$sig_star <- ""
sgt$sig_star[sgt$coefficients > 0 & sgt$lower > 0] <- "*"
sgt$sig_star[sgt$coefficients < 0 & sgt$upper < 0] <- "*"
sgt$coefficients <- format(round(sgt$coefficients, digits = 2),
  digits = 1, nsmall = 2)
sgt$lower <- format(round(sgt$lower, digits = 2),
  digits = 1, nsmall = 2)
sgt$upper <- format(round(sgt$upper, digits = 2),
  digits = 1, nsmall = 2)
sgt$prb <- NA
sgt$coef <- paste(sgt$coefficients,
  " (", sgt$lower, " to ", sgt$upper,")", sgt$sig_star, sep = '')

sgt_agg <- filter(sgt, standardization == "None" & variable == "Aggression")
sgt_act <- filter(sgt, standardization == "None" & variable == "Activity")
sgt_doc <- filter(sgt, standardization == "None" & variable == "Docility")

sgt_agg_sd <- filter(sgt, standardization == "SD" &  variable == "Aggression")
sgt_act_sd <- filter(sgt, standardization == "SD" &  variable == "Activity")
sgt_doc_sd <- filter(sgt, standardization == "SD" &  variable == "Docility")

sgt_doc_ms <- filter(sgt, standardization == "Mean" &  variable == "Docility")

N <- fit_raneff_data %>%
  filter(type == "blup") %>%
  group_by(Year, add = FALSE) %>%
  summarise(n = n(), t_kprod = sum(kprod), t_ars = sum(ars_all), mean_docil = mean(docility, na.rm =

pandoc.table(
  data.frame(
    Year = N$Year,
    Aggression = sgt_agg$coef,
    Activity = sgt_act$coef,
    Docility = sgt_doc$coef
  ),
  caption ="Traditional selection gradients (ignoring behavioural uncertainty). Not standardized."
)
```

Year	Aggression	Activity
2003	-0.26 (-1.66 to 0.61)	0.90 (-0.25 to 1.76)
2004	-0.75 (-1.36 to -0.05)*	0.57 (-0.18 to 1.35)

Year	Aggression	Activity
2005	-0.56 (-1.02 to -0.06)*	0.46 (-0.08 to 0.98)
2006	0.98 (0.20 to 1.86)*	-0.33 (-1.11 to 0.48)
2007	-0.09 (-0.92 to 0.49)	0.19 (-0.53 to 0.88)
2008	1.11 (-0.36 to 2.30)	-0.60 (-1.46 to 0.94)
2009	0.75 (-0.09 to 1.39)	-1.38 (-2.11 to -0.60)*
2010	-0.03 (-0.74 to 0.67)	0.12 (-0.54 to 0.77)

Table 29: Traditional selection gradients (ignoring behavioural uncertainty). Not standardized. (continued below)

Docility
-0.11 (-0.36 to 0.10)
0.07 (-0.13 to 0.24)
0.05 (-0.04 to 0.16)
-0.10 (-0.22 to 0.04)
0.01 (-0.11 to 0.13)
-0.04 (-0.21 to 0.13)
-0.12 (-0.23 to 0.00)*
-0.05 (-0.12 to 0.04)

```
pandoc.table(
  data.frame(
    Year = N$Year,
    Aggression = sgt_agg_sd$coef,
    Activity = sgt_act_sd$coef,
    Docility = sgt_doc_sd$coef
  ),
  caption ="Traditional selection gradients (ignoring behavioural uncertainty). SD-standardized."
)
```

Year	Aggression	Activity
2003	-0.17 (-1.25 to 0.57)	0.86 (-0.20 to 1.96)
2004	-0.62 (-1.17 to 0.22)	0.49 (-0.37 to 1.29)
2005	-0.41 (-0.71 to -0.04)*	0.34 (-0.10 to 0.76)
2006	0.66 (0.13 to 1.19)*	-0.23 (-0.96 to 0.35)
2007	-0.07 (-0.57 to 0.48)	0.15 (-0.44 to 0.63)
2008	0.78 (0.20 to 1.54)*	-0.44 (-1.64 to 0.39)

Year	Aggression	Activity
2009	0.56 (0.05 to 1.20)*	-1.01 (-1.84 to -0.47)*
2010	-0.02 (-0.43 to 0.56)	0.10 (-0.34 to 0.50)

Table 31: Traditional selection gradients (ignoring behavioural uncertainty). SD-standardized. (continued below)

Docility
-0.40 (-1.22 to 0.61)
0.26 (-0.29 to 1.01)
0.19 (-0.11 to 0.54)
-0.41 (-0.92 to 0.17)
0.03 (-0.47 to 0.52)
-0.17 (-0.81 to 0.45)
-0.56 (-1.07 to -0.06)*
-0.21 (-0.47 to 0.16)

```
pandoc.table(
  data.frame(
    Year = N$Year,
    Docility = sgt_doc_ms$coefficients,
    mean = N$mean_docil
  ),
  caption ="Traditional selection gradients (ignoring behavioural uncertainty). Mean-standardized."
)
```

Year	Docility	mean
2003	-1.86	17.29
2004	1.25	17.31
2005	0.88	17.25
2006	-1.72	16.77
2007	0.12	16.77
2008	-0.63	16.9
2009	-2.04	16.79
2010	-0.79	17.19

Table 33: Traditional selection gradients (ignoring behavioural uncertainty). Mean-standardized.

Plot

Female Linear Selection Gradients ARS

Linear selection gradients ± 95% credible intervals for female behavioral traits on annual reproductive success.

```
load("data/analyses_data/sel_grads_blup.RData")
sel_grads_blup$post_mode <- sel_grads_blup$coefficients
sel_grads_blup$upper_sig_star <- ""
sel_grads_blup$lower_sig_star <- ""
sel_grads_blup$upper_sig_star[sel_grads_blup$coefficients > 0 &
  sel_grads_blup$lower > 0] <- "*"
sel_grads_blup$lower_sig_star[sel_grads_blup$coefficients < 0 &
  sel_grads_blup$upper < 0] <- "*"
sel_grads_blup$upper_sig_01_star <- ""
sel_grads_blup$lower_sig_01_star <- ""
sel_grads_blup$upper_sig_01_star[sel_grads_blup$coefficients > 0
  & sel_grads_blup$lower_1 > 0] <- "."
sel_grads_blup$lower_sig_01_star[sel_grads_blup$coefficients < 0
  & sel_grads_blup$upper_1 < 0] <- "."
sel_grads_blup$upper_sig_01_star[sel_grads_blup$coefficients > 0
  & sel_grads_blup$lower > 0] <- ""
sel_grads_blup$lower_sig_01_star[sel_grads_blup$coefficients < 0
  & sel_grads_blup$upper < 0] <- ""

pdf(file = "figure/04_sg_blup_SD_print.pdf", width = 4.33, height = 3)
p <- p_sel_grad_MCMC %+% filter(sel_grads_blup, standardization == "SD")
p <- p + ylab("Coefficient ± 0.95 Confidence Interval")
p <- p + geom_text(aes(x = Year, y = upper, group = variable,
  label = upper_sig_01_star), vjust = -0.3,
  position = position_dodge(width = 0.5), size = 7)
p <- p + geom_text(aes(x = Year, y = lower, group = variable,
  label = lower_sig_01_star), vjust =  0.5,
  position = position_dodge(width = 0.5), size = 7)
p <- p + geom_text(aes(x = Year, y = upper, group = variable,
  label = upper_sig_star), vjust = -0.3,
  position = position_dodge(width = 0.5), size = 5)
p <- p + geom_text(aes(x = Year, y = lower, group = variable,
  label = lower_sig_star), vjust =  1.3,
  position = position_dodge(width = 0.5), size = 5)
p <- p + ylim(c(-2, 2))
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will replace the existing scale.
```

```
p <- p + scale_shape_manual(name = "A", values = c(24, 21, 22))
```

```
## Scale for 'shape' is already present. Adding another scale for 'shape', which will replace the existing
```

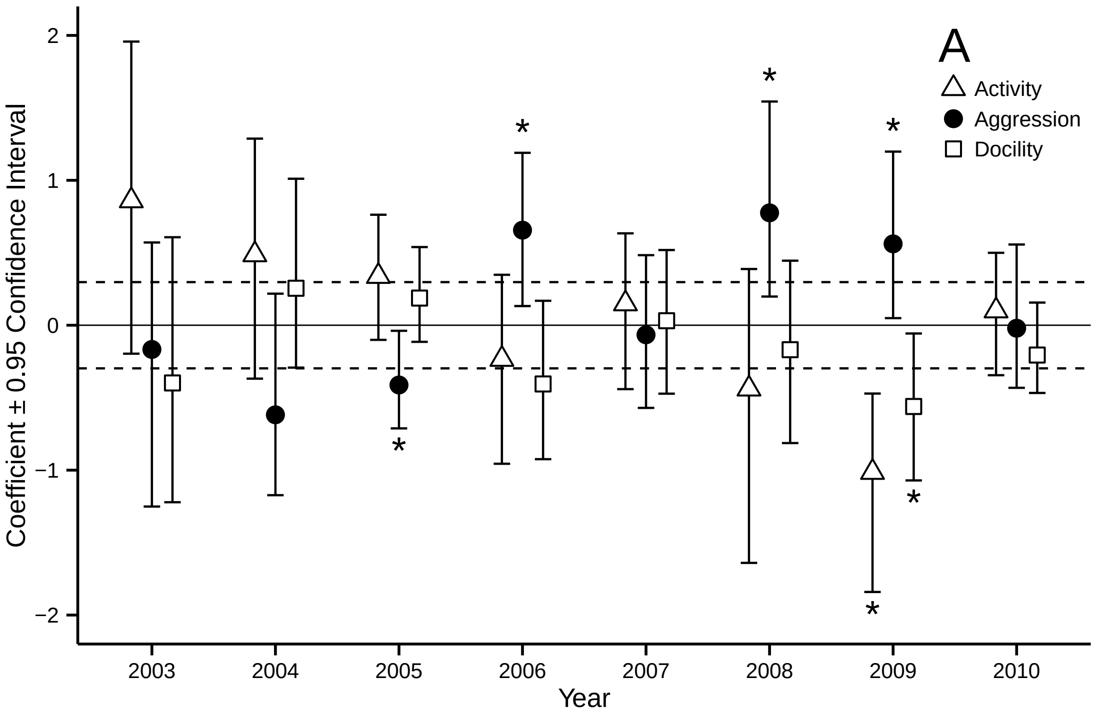
```
p <- p + scale_fill_manual(name = "A", values = c("white", "black", "white"))
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which will replace the existing s
```

```
p <- p + scale_color_manual(name = "A", values = c("black", "black", "black"))

## Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the exist

p

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

dev.off()

## pdf
##   2

p

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```

Correlations

```
load("data/analyses_data/sel_grads_blup.RData")
sel_grads_blup_flat <- data.frame(
  Aggression = filter(sel_grads_blup, standardization == "SD" &
   variable == "Aggression")$coefficients,
  Agg_upper  = filter(sel_grads_blup, standardization == "SD" &
   variable == "Aggression")$upper,
  Agg_lower  = filter(sel_grads_blup, standardization == "SD" &
   variable == "Aggression")$lower,
  Activity   = filter(sel_grads_blup, standardization == "SD" &
   variable == "Activity")$coefficients,
  Act_upper  = filter(sel_grads_blup, standardization == "SD" &
   variable == "Activity")$upper,
  Act_lower  = filter(sel_grads_blup, standardization == "SD" &
   variable == "Activity")$lower,
  Docility   = filter(sel_grads_blup, standardization == "SD" &
   variable == "Docility")$coefficients,
  Doc_upper  = filter(sel_grads_blup, standardization == "SD" &
```

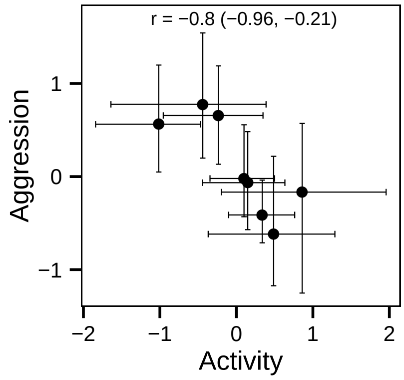



```
    variable == "Docility")$upper,
  Doc_lower  = filter(sel_grads_blup, standardization == "SD" &
    variable == "Docility")$lower
)

cor.behav <- function(x, y){
  ct <- cor.test(x, y)
  out <- data.frame(est = ct$estimate, lower = ct$conf.int[1],
    upper = ct$conf.int[2], stringsAsFactors = FALSE)
  out <- round(out, digits = 2)
  out$print <- paste(out$est, " (", out$lower, ", ", out$upper, ")", sep = "")
}

cor_blup_agg_act <- cor.behav(sel_grads_blup_flat$Aggression,
  sel_grads_blup_flat$Activity)
cor_blup_agg_doc <- cor.behav(sel_grads_blup_flat$Aggression,
  sel_grads_blup_flat$Docility)
cor_blup_doc_act <- cor.behav(sel_grads_blup_flat$Docility,
  sel_grads_blup_flat$Activity)
```

Aggression and Activity

```
p <- ggplot(data = sel_grads_blup_flat, aes(x = Activity, y = Aggression))
p <- p + geom_point()
p <- p + ylab("Aggression")
```

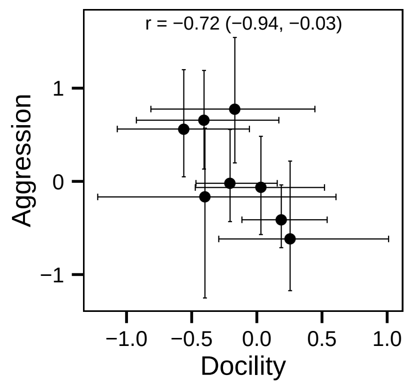
```
p <- p + xlab("Activity")
p <- p + theme_bw(base_size = 10)
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + geom_errorbarh(aes(xmin = Act_lower, xmax = Act_upper),
  height = 0.07, size = 0.2)
p <- p + geom_errorbar(aes(ymin = Agg_lower, ymax = Agg_upper),
  width = 0.07, size = 0.2)
p <- p + annotate(geom = "text", size = 2.5, x = 0.1, y = 1.7,
  label = paste("r = ", cor_blup_agg_act, sep = ''))
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



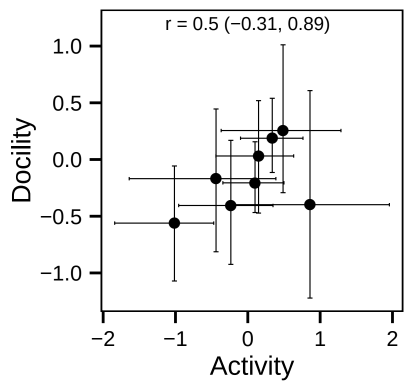
Aggression and Docility

```
p <- ggplot(data = sel_grads_blup_flat, aes(y = Aggression, x = Docility))
p <- p + geom_point()
p <- p + ylab("Aggression")
p <- p + xlab("Docility")
p <- p + theme_bw(base_size = 10)
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + geom_errorbarh(aes(xmin = Doc_lower, xmax = Doc_upper),
  height = 0.07, size = 0.2)
p <- p + geom_errorbar(aes(ymin = Agg_lower, ymax = Agg_upper),
  width = 0.03, size = 0.2)
p <- p + annotate(geom = "text", size = 2.5, x = -0.1, y = 1.7,
  label = paste("r = ", cor_blup_agg_doc, sep = ''))
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```

Activity and Docility




```
p <- ggplot(data = sel_grads_blup_flat, aes(x = Activity, y = Docility))
p <- p + geom_point()
p <- p + xlab("Activity")
p <- p + ylab("Docility")
p <- p + theme_bw(base_size = 10)
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + geom_errorbar(aes(ymin = Doc_lower, ymax = Doc_upper),
  width = 0.07, size = 0.2)
p <- p + geom_errorbarh(aes(xmin = Act_lower, xmax = Act_upper),
  height = 0.03, size = 0.2)
p <- p + annotate(geom = "text", size = 2.5, x = 0, y = 1.2,
  label = paste("r = ", cor_blup_doc_act, sep = ''))
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Summary Statistics

```
library(dplyr)
sign_change <- function(x){
  # sign changes calculated as the number of changes in direction between
  # successive years relative to n-1
```

```
  s <- sign(x)
  sum(s[1:(length(s)-1)] != s[-1])/(length(s)-1)
}
se <- function(lower, upper){
  (upper - lower) / 3.92
}

sum_stats_blup <- sel_grads_blup %>%
filter(standardization == "SD") %>%
group_by(variable, add = FALSE) %>%
summarise(
  mean_abs_b = mean(abs(coefficients)),
  abs_mean_b = abs(mean(coefficients)),
  sd_b = sd(coefficients),
  mean_se_b = mean(se(lower, upper)),
  freq_sign = sign_change(coefficients),
  mean_cv = mean(se(lower, upper) / abs(coefficients))
  )
sum_stats_blup[ ,2:6] <- round(sum_stats_blup[ ,2:6], 2)
pandoc.table(sum_stats_blup)
```

variable	mean_abs_b	abs_mean_b	sd_b	mean_se_b
Activity	0.45	0.03	0.58	0.36
Aggression	0.41	0.09	0.52	0.3
Docility	0.28	0.16	0.3	0.28

Table 34: Table continues below

freq_sign	mean_cv
0.57	1.138
0.57	2.701
0.57	1.912

Compare Analytical Frameworks

Table

```
load("data/analyses_data/sel_grads_blup.RData")
load("data/analyses_data/sel_grads_mcmc.RData")

sg_blups <- sel_grads_blup %>% filter(standardization == "SD")
sg_mcmc <- sel_grads_mcmc %>% filter(standardization == "SD")

compare_grads <- left_join(select(sg_blups, Year, variable,
```

```
  blup_coef = coefficients, blup_upper = upper, blup_lower = lower),
  select(sg_mcmc, Year, variable, mcmc_pm = post_mode,
    mcmc_upper = upper, mcmc_lower = lower), by = c("variable", "Year"))

ct_print <- function(x,y){
  ct <- cor.test(x,y)
  est <- format(ct$estimate, digits = 2)
  ci <- format(ct$conf.int, digits = 2)
  ct <- format(ct, digits = 2)
  paste(est, " (", ci[1], ", ", ci[2] ,")", sep = '')
}

c_table <- compare_grads %>%
  group_by(variable) %>%
  summarise(cor = cor(blup_coef, mcmc_pm),
    abs_diff = mean((abs(blup_coef - mcmc_pm))),
    mean_mcmc = mean(abs(mcmc_pm)), mean_blup = mean(abs(blup_coef)),
    prop_diff = mean_blup / mean_mcmc, cor_test = ct_print(blup_coef, mcmc_pm),
    lmerGreater = sum(abs(blup_coef) > abs(mcmc_pm))
  )
pandoc.table(c_table)
```

variable	cor	abs_diff	mean_mcmc	mean_blup
Activity	0.8931	0.2345	0.2374	0.4527
Aggression	0.9599	0.2312	0.1904	0.4096
Docility	0.9489	0.08657	0.1957	0.2766

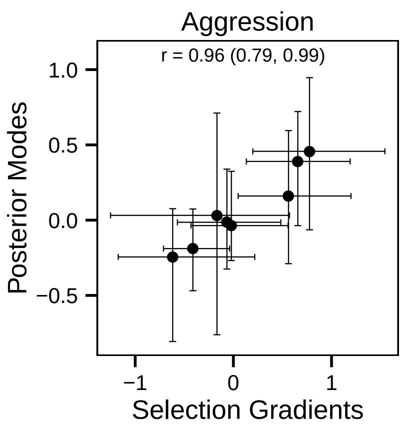
Table 36: Table continues below

prop_diff	cor_test	lmerGreater
1.907	0.89 (0.51, 0.98)	8
2.152	0.96 (0.79, 0.99)	7
1.413	0.95 (0.74, 0.99)	7

Aggression plot

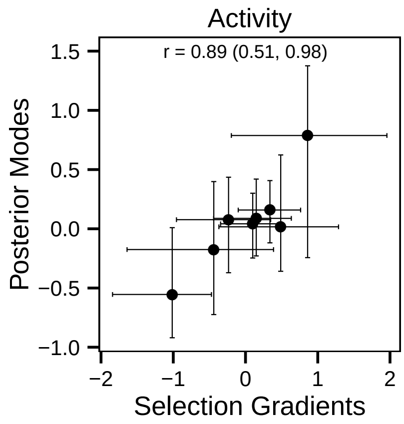
```
p <- ggplot(filter(compare_grads, variable == "Aggression"),
  aes(x = blup_coef, y = mcmc_pm))
p <- p + geom_point()
p <- p + geom_errorbarh(aes(xmin = blup_lower, xmax = blup_upper),
  height = 0.04, size = 0.2)
p <- p + geom_errorbar(aes(ymin = mcmc_lower, ymax = mcmc_upper),
  width = 0.07, size = 0.2)
p <- p + theme_bw(base_size = 10)
```

```
p <- p + theme(plot.title = element_text(size = 10),
  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  panel.background = element_blank(), strip.background = element_blank(),
  strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + annotate(geom = "text", size = 2.5, x = 0.1, y = 1.1,
  label = paste("r = ", filter(c_table, variable == "Aggression") %>%
  select(cor_test), sep = ''))
p <- p + ylab("Posterior Modes")
p <- p + xlab("Selection Gradients")
p <- p + ggtitle("Aggression")
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Activity Plot

```
p <- ggplot(filter(compare_grads, variable == "Activity"),
  aes(x = blup_coef, y = mcmc_pm))
p <- p + geom_point()
p <- p + geom_errorbarh(aes(xmin = blup_lower, xmax = blup_upper),
  height = 0.04, size = 0.2)
p <- p + geom_errorbar(aes(ymin = mcmc_lower, ymax = mcmc_upper),
  width = 0.07, size = 0.2)
p <- p + theme_bw(base_size = 10)
p <- p + theme(plot.title = element_text(size = 10),
  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  panel.background = element_blank(), strip.background = element_blank(),
  strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + annotate(geom = "text", size = 2.5, x = 0, y = 1.5,
  label = paste("r = ", filter(c_table, variable == "Activity") %>%
  select(cor_test), sep = ''))
p <- p + ylab("Posterior Modes")
p <- p + xlab("Selection Gradients")
p <- p + ggtitle("Activity")
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



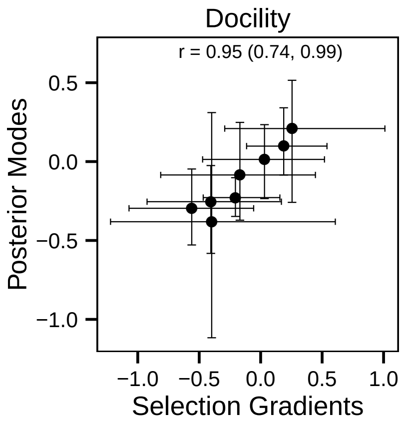
Docility Plot

```
p <- ggplot(filter(compare_grads, variable == "Docility"),
  aes(x = blup_coef, y = mcmc_pm))
p <- p + geom_point()
p <- p + geom_errorbarh(aes(xmin = blup_lower, xmax = blup_upper),
  height = 0.04, size = 0.2)
p <- p + geom_errorbar(aes(ymin = mcmc_lower, ymax = mcmc_upper),
  width = 0.07, size = 0.2)
p <- p + theme_bw(base_size = 10)
p <- p + theme(plot.title = element_text(size = 10),
  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  panel.background = element_blank(), strip.background = element_blank(),
  strip.text = element_blank(),
  panel.border = element_rect(linetype = "solid", colour = "black"))
p <- p + ylab("Posterior Modes")
p <- p + xlab("Selection Gradients")
p <- p + annotate(geom = "text", size = 2.5, x = 0, y = 0.7,
  label = paste("r = ", filter(c_table, variable == "Docility") %>%
  select(cor_test), sep = ''))
p <- p + ggtitle("Docility")
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```

Competition and Selection

Accounting for behavioral measurement uncertainty

```
library(MASS) # MASS clashes with dplyr... so always load first
library(pander) # pander clashes with dplyr... so always load first
library(ggplot2)
library(grid)
library(dplyr)
set.alignment('right', row.names = 'left')
load("data/analyses_data/sel_grads_mcmc.RData")
```




```
fitness <- read.table(file = "data/fitness+competition.csv", sep = ',',
  header = TRUE, stringsAsFactors = FALSE)
load("data/analyses_data/fit_raneff_data.RData")
```

We will examine the effect of competition on selection in two general steps.

1. Is there an interaction between competition and behavior on fitness?
2. Are there nonlinear effects of behavior on fitness?

Correlations between selection gradients and competition

First a plot of the relationship between selection gradients and competition. The two study areas were pooled to calculate selection gradients for each year. Therefore we need to calculate competition for the combined study areas. Competition is the number of offspring produced during the year divided by the number of offspring that survived to spring (i.e. recruited into the population).

```
competition_year <- fitness %>%
  filter(grid_year != "SU2008") %>%
  select(Year, competition) %>%
  unique() %>%
  group_by(Year, add = FALSE) %>%
  summarise(mean_competition = mean(competition))

n_year <- filter(fit_raneff_data, type == "blup") %>%
  group_by(Year, add = FALSE) %>% summarise(n = n())
competition_year <- left_join(competition_year, n_year, by = "Year")
competition_year$Year <- as.character(competition_year$Year)

load("data/analyses_data/sel_grads_mcmc.RData")
sel_grads_mcmc_comp <- left_join(
  filter(sel_grads_mcmc, standardization == "SD"), competition_year,
  by = "Year")
save(sel_grads_mcmc_comp, competition_year,
  file = "data/analyses_data/sel_grads_mcmc_comp.RData")
```

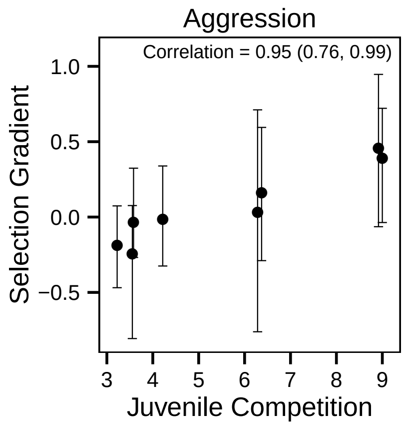
```r
load("data/analyses_data/sel_grads_mcmc_comp.RData")

cor_sgrad_comp <- function(x){
  v <- x$variable[1]
  ct <- cor.test(x$post_mode, x$mean_competition)
  data.frame(variable = v, est = ct$estimate, lower = ct$conf.int[1],
    upper = ct$conf.int[2], stringsAsFactors = FALSE)
}

mcmc_cor <- sel_grads_mcmc_comp %>%
  group_by(variable, add = FALSE) %>%
  do(cor_sgrad_comp(.))
mcmc_cor[ ,2:4] <- round(mcmc_cor[ ,2:4], digits = 2)
mcmc_cor$print <- paste(mcmc_cor$est,
  " (", mcmc_cor$lower, ", ", mcmc_cor$upper, ")", sep = "")
```

Aggression and Competition

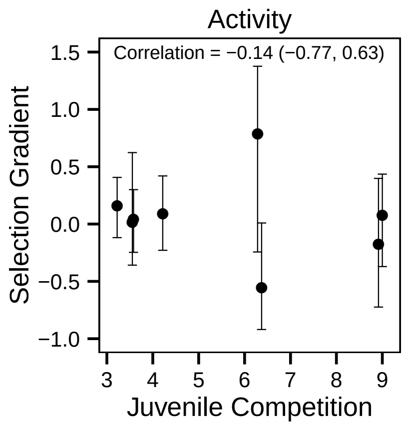
```r
p <- ggplot(data = filter(sel_grads_mcmc_comp, variable == "Aggression"),
  aes(x = mean_competition, y = post_mode))
p <- p + geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, size = 0.2)
p <- p + geom_point()
p <- p + theme_bw(base_size = 10)
p <- p + scale_x_continuous(breaks = c(3,4,5,6,7,8,9))
p <- p + ylab("Selection Gradient")
p <- p + ggtitle("Aggression")
p <- p + xlab("Juvenile Competition")
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_text(size = 10),
  panel.border = element_rect(linetype = "solid", colour = "black"),
  plot.title = element_text(size = 10))
p <- p + geom_text(data = filter(mcmc_cor, variable == "Aggression"),
  aes(x = 6.5, y = 1.1, label = paste("Correlation = ", print, sep = '')),
  size = 2.5)
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```

Activity and Competition

```r
p <- ggplot(data = filter(sel_grads_mcmc_comp, variable == "Activity"),
  aes(x = mean_competition, y = post_mode))
p <- p + geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, size = 0.2)
p <- p + geom_point()
p <- p + theme_bw(base_size = 10)
p <- p + scale_x_continuous(breaks = c(3,4,5,6,7,8,9))
p <- p + ylab("Selection Gradient")
p <- p + ggtitle("Activity")
p <- p + xlab("Juvenile Competition")
p <- p + theme(panel.grid.major = element_blank(),
```



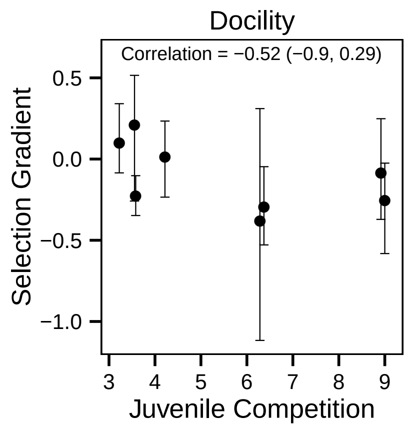

```
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_text(size = 10),
  panel.border = element_rect(linetype = "solid", colour = "black"),
  plot.title = element_text(size = 10))
p <- p + ylim(c(-1, 1.5))
p <- p + geom_text(data = filter(mcmc_cor, variable == "Activity"),
  aes(x = 6.1, y = 1.5, label = paste("Correlation = ", print, sep = '')),
  size = 2.5)
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Docility and Competition

```
p <- ggplot(data = filter(sel_grads_mcmc_comp, variable == "Docility"),
  aes(x = mean_competition, y = post_mode))
p <- p + geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, size = 0.2)
p <- p + geom_point()
p <- p + theme_bw(base_size = 10)
p <- p + scale_x_continuous(breaks = c(3,4,5,6,7,8,9))
p <- p + ylab("Selection Gradient")
p <- p + ggtitle("Docility")
```

```
p <- p + xlab("Juvenile Competition")
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_text(size = 10),
  panel.border = element_rect(linetype = "solid", colour = "black"),
  plot.title = element_text(size = 10))
p <- p + geom_text(data = filter(mcmc_cor, variable == "Docility"),
  aes(x = 6.1, y = 0.65, label = paste("Correlation = ", print, sep = '')),
  size = 2.5)
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Effect of competition on linear selection (glmms)

```
load("data/analyses_data/fit_raneff_data.RData")
library(dplyr)
fit_raneff_data <- tbl_df(fit_raneff_data)
library(lme4)

# Model with interactions between competition and the behavioral traits.
# grid_year and ID are random effects.
arsLinearCompetition <- function(dat){
  ars_linear_comp <- glmer(
    ars_all ~
    competition_s +
    aggression_s +
    competition_s:aggression_s +
    activity_s +
    competition_s:activity_s +
    docility_s +
    competition_s:docility_s +
    (1|Grid) + (1|ID),
    data = dat, family = poisson, control=glmerControl(optimizer="bobyqa")
    )
  random_effect_variances <- VarCorr(ars_linear_comp)
  data.frame(t(summary(ars_linear_comp)$coefficients[ ,"Estimate"]),
```

```
  ID = random_effect_variances$ID[1], Grid = random_effect_variances$Grid[1])
}

library(foreach)

## foreach: simple, scalable parallel programming from Revolution Analytics
## Use Revolution R for scalability, fault tolerance and more.
## http://www.revolutionanalytics.com

library(doMC)

## Loading required package: iterators
## Loading required package: parallel

ncores = 12
registerDoMC(cores = ncores)

batches <- data.frame(start = seq(1, 1000, round(1000/ncores))[1:ncores])
batches$stop <- c(batches$start[2:length(batches$start)] - 1, 1000)

start_time <- Sys.time()
ars_linear_comp_posterior <- foreach(i = 1:ncores, .combine = rbind) %dopar% {
  results <- fit_raneff_data %>%
    filter(type == "raneff", itt %in% batches$start[i]:batches$stop[i]) %>%
    group_by(itt, add = FALSE) %>%
    do(arsLinearCompetition(.))
}
run_time <- Sys.time() - start_time
print(run_time)

## Time difference of 2.539 mins

save(ars_linear_comp_posterior,
  file = "data/analyses_data/ars_linear_comp_posterior.RData")

load("data/analyses_data/ars_linear_comp_posterior.RData")
library(MCMCglmm)
library(lme4)
library(data.table)

## data.table 1.9.2  For help type: help("data.table")
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:dplyr':
##
##     between, last
```

```
getPosteriorParams <- function(x){
  require(MCMCglmm)
  dat_mcmc <- mcmc(x)
  pm <- posterior.mode(dat_mcmc)
  hpd <- HPDinterval(dat_mcmc, prob = 0.9)
  pm_table <-  format(round(pm , digits = 2), digits = 1, nsmall = 2,
    scientific = FALSE)
  hpd_table <- format(round(hpd, digits = 2), digits = 1, nsmall = 2,
    scientific = FALSE)
  pm_hpd_table <- data.frame(cbind(pm_table, hpd_table))
  pm_hpd_table$pm_hpd <- paste(
    pm_table, " (", hpd_table[ ,1], ", ", hpd_table[ ,2], ")", sep = ''
    )
  pm_hpd_table$sig[sign(hpd[ ,1]) == sign(hpd[ ,2])] <- "*"
  pm_hpd_table$sig[sign(hpd[ ,1]) != sign(hpd[ ,2])] <- " "
  pm_hpd_table$space[sign(hpd[ ,1]) == sign(hpd[ ,2])] <- "*"
  pm_hpd_table$space[sign(hpd[ ,1]) != sign(hpd[ ,2])] <- " "
  pm_hpd_table$pm_hpd <- paste(pm_hpd_table$pm_hpd, pm_hpd_table$sig, sep = "")
  return(pm_hpd_table)
}

linear_hpd_ars <- getPosteriorParams(ars_linear_comp_posterior %>%
  ungroup() %>%
  select(Intercept = X.Intercept., Competition = competition_s,
    Aggression = aggression_s, Activity = activity_s, docility = docility_s,
    "Competition x Aggression" = competition_s.aggression_s,
    "Competition x Activity" = competition_s.activity_s,
    "Competition x Docility" = competition_s.docility_s)
    )
```

Interaction between competition and linear selection results The effect of competition on linear selection on female behavioral traits for annual reproductive success. Posterior modes are given with highest posterior density intervals in parentheses.

```
pandoc.table(linear_hpd_ars %>% select(pm_hpd), justify="right")
```

	pm_hpd
Intercept	-1.01 (-1.11, -0.93)*
Competition	-3.27 (-3.65, -3.09)*
Aggression	0.36 (0.13, 0.63)*
Activity	-0.12 (-0.37, 0.15)
docility	-0.18 (-0.38, -0.06)*
Competition x Aggression	1.30 (0.50, 1.99)*
Competition x Activity	-0.45 (-1.26, 0.32)
Competition x Docility	-0.54 (-1.13, -0.09)*

Competition and nonlinear selection (glmms)

```
load("data/analyses_data/fit_raneff_data.RData")
library(lme4)

arsNonlinearResults <- function(dat){
  ars_model  <- glmer(ars_all ~ aggression_s*competition_s +
    activity_s*competition_s + docility_s*competition_s +
    aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
    I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
    (1 | grid_year) + (1|ID), data = dat, family = poisson,
    control=glmerControl(optimizer="bobyqa"))
  kpd_model  <- glmer(kprod   ~ aggression_s*competition_s +
    activity_s*competition_s + docility_s*competition_s +
    aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
    I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
    (1 | grid_year) + (1|ID), data = dat, family = poisson,
    control=glmerControl(optimizer="bobyqa"))
  ows_model  <- glmer(prop    ~ aggression_s*competition_s +
    activity_s*competition_s + docility_s*competition_s +
    aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
    I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
    (1 | grid_year) + (1|ID), data = dat, weights = kprod, family = binomial,
    control=glmerControl(optimizer="bobyqa"))

  ars_vc <- VarCorr(ars_model)
  ars_t <- data.table(fitness = "ars",
    t(summary(ars_model)$coefficients[ ,"Estimate"]), ID = ars_vc$ID[1],
    grid_year = ars_vc$grid_year[1])
  kpd_vc <- VarCorr(kpd_model)
  kpd_t <- data.table(fitness = "kpd",
    t(summary(kpd_model)$coefficients[ ,"Estimate"]), ID = kpd_vc$ID[1],
    grid_year = kpd_vc$grid_year[1])
  ows_vc <- VarCorr(ows_model)
  ows_t <- data.table(fitness = "ows",
    t(summary(ows_model)$coefficients[ ,"Estimate"]), ID = ows_vc$ID[1],
    grid_year = ows_vc$grid_year[1])
  rbind(rbind(ars_t, kpd_t), ows_t)
}


library(foreach)
library(doMC)
ncores = 12
registerDoMC(cores = ncores)

batches <- data.frame(start = seq(1, 1000, round(1000/ncores))[1:ncores])
batches$stop <- c(batches$start[2:length(batches$start)] - 1, 1000)

start_time <- Sys.time()
nonlinear_mcmc <- foreach(i = 1:ncores, .combine = rbind) %dopar% {
```

```
  results <- fit_raneff_data %>%
    filter(type == "raneff", itt %in% batches$start[i]:batches$stop[i]) %>%
    group_by(itt, add = FALSE) %>%
    do(arsNonlinearResults(.))
}
run_time <- Sys.time() - start_time
print(run_time)
```

```
## Time difference of 20.4 mins
```

```
save(nonlinear_mcmc, file = "data/analyses_data/nonlinear_mcmc_models.RData")
```

```
load("data/analyses_data/nonlinear_mcmc_models.RData")

pm_hpd_ars <- getPosteriorParams(
  nonlinear_mcmc[nonlinear_mcmc$fitness =="ars", 3:18])
pm_hpd_ows <- getPosteriorParams(
  nonlinear_mcmc[nonlinear_mcmc$fitness =="ows", 3:18])
pm_hpd_kpd <- getPosteriorParams(
  nonlinear_mcmc[nonlinear_mcmc$fitness =="kpd", 3:18])

nonlinear_results_mcmc <- data.frame(ARS = pm_hpd_ars$pm_hpd,
  OWS = pm_hpd_ows$pm_hpd, Fecundity = pm_hpd_kpd$pm_hpd)

row.names(nonlinear_results_mcmc) <- c("Intercept", "Aggression",
 "Competition", "Activity", "Docility", "Aggression^2", "Activity^2",
 "Docility^2", "Aggression x Competition", "Activity x Competition",
 "Docility x Competition", "Aggression x Activity",
 "Aggression^2 x Competition", "Activity^2 x Competition",
 "Docility^2 x Competition", "Agg. x Act. x Competition"
 )
```

Nonlinear results

The effect of competition on linear and nonlinear selection on female behavioral traits for annual reproductive success. Posterior modes are given with highest posterior density intervals in parentheses.

```
pandoc.table(nonlinear_results_mcmc[c(1,3,2,4:16), ],
  split.tables = 160)
```

	ARS	OWS	Fecundity
Intercept	-0.76 (-1.18, -0.42)*	-1.84 (-2.35, -1.42)*	1.34 (1.28, 1.41)*
Competition	-2.58 (-3.98, -1.72)*	-2.41 (-4.21, -1.27)*	-0.06 (-0.19, 0.09)
Aggression	0.45 (0.08, 0.82)*	0.59 (0.17, 1.13)*	-0.04 (-0.09, 0.02)
Activity	-0.17 (-0.53, 0.17)	-0.06 (-0.68, 0.24)	0.03 (-0.05, 0.06)
Docility	-0.29 (-0.55, -0.13)*	-0.29 (-0.54, -0.03)*	0.04 (0.00, 0.07)*

	ARS	OWS	Fecundity
Aggression^2	-0.12 (-0.51, 0.15)	-0.28 (-0.70, 0.16)	0.02 (-0.05, 0.05)
Activity^2	0.00 (-0.36, 0.23)	-0.07 (-0.42, 0.31)	-0.02 (-0.07, 0.02)
Docility^2	-0.15 (-0.33, 0.04)	-0.20 (-0.41, 0.05)	0.01 (-0.03, 0.04)
Aggression x Competition	1.71 (0.48, 2.66)*	2.16 (0.55, 3.53)*	-0.03 (-0.16, 0.09)
Activity x Competition	-0.72 (-1.77, 0.35)	-0.24 (-1.70, 1.10)	-0.03 (-0.15, 0.08)
Docility x Competition	-0.70 (-1.55, -0.26)*	-0.45 (-1.13, 0.49)	-0.04 (-0.14, 0.02)
Aggression x Activity	0.14 (-0.22, 0.67)	0.21 (-0.29, 0.84)	0.02 (-0.06, 0.08)
Aggression^2 x Competition	-0.27 (-1.36, 0.56)	-0.35 (-1.98, 0.70)	0.03 (-0.11, 0.14)
Activity^2 x Competition	-0.06 (-1.07, 0.68)	-0.11 (-1.32, 1.00)	0.00 (-0.11, 0.11)
Docility^2 x Competition	-0.51 (-1.05, 0.15)	-0.34 (-1.29, 0.19)	-0.02 (-0.11, 0.05)
Agg. x Act. x Competition	0.44 (-0.75, 1.92)	0.52 (-1.08, 2.57)	0.01 (-0.20, 0.16)

Ignoring behavioural uncertainty

```
load("data/analyses_data/sel_grads_blup.RData")
load("data/analyses_data/fit_raneff_data.RData")
```

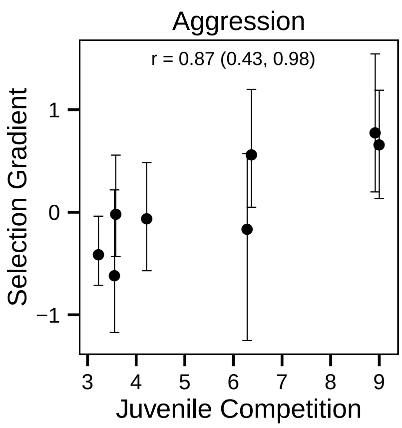
Correlations between selection gradients and competition

```
competition_year <- fitness %>%
  filter(grid_year != "SU2008") %>%
  select(Year, competition) %>%
  unique() %>%
  group_by(Year, add = FALSE) %>%
  summarise(mean_competition = mean(competition))

n_year <- filter(fit_raneff_data, type == "blup") %>%
  group_by(Year, add = FALSE) %>%
  summarise(n = n())
competition_year <- left_join(competition_year, n_year, by = "Year")
competition_year$Year <- as.character(competition_year$Year)

load("data/analyses_data/sel_grads_blup.RData")
sel_grads_blup_competition <- left_join(
  filter(sel_grads_blup, standardization == "SD"), competition_year,
  by = "Year")
save(sel_grads_blup_competition, competition_year,
  file = "data/analyses_data/sel_grads_blup_competition.RData")

load("data/analyses_data/sel_grads_blup_competition.RData")

cor_sgrad_comp <- function(x){
```

```
  v <- x$variable[1]
  ct <- cor.test(x$coefficients, x$mean_competition)
  data.frame(variable = v, est = ct$estimate, lower = ct$conf.int[1],
    upper = ct$conf.int[2], stringsAsFactors = FALSE)
}

sg.comp <- sel_grads_blup_competition %>%
  group_by(variable, add = FALSE) %>%
  do(cor_sgrad_comp(x=.))
sg.comp[ ,2:4] <- round(sg.comp[ ,2:4], digits = 2)
sg.comp$print <- paste(sg.comp$est, " (", sg.comp$lower, ", ",
  sg.comp$upper, ")", sep = "")
```

Aggression and Competition

```
p <- ggplot(data = filter(sel_grads_blup_competition,
  variable == "Aggression"), aes(x = mean_competition, y = coefficients))
p <- p + geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, size = 0.2)
p <- p + geom_point()
p <- p + theme_bw(base_size = 10)
p <- p + scale_x_continuous(breaks = c(3,4,5,6,7,8,9))
p <- p + ylab("Selection Gradient")
p <- p + ggtitle("Aggression")
p <- p + xlab("Juvenile Competition")
p <- p + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.backgro
p <- p + geom_text(data = filter(sg.comp, variable == "Aggression"),
  aes(x = 6, y = 1.5, label = paste("r = ", print, sep = '')), size = 2.5)
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



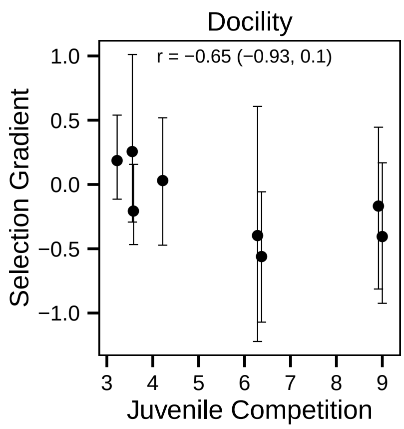
Activity and Competition

```
p <- ggplot(data = filter(sel_grads_blup_competition, variable == "Activity"),
  aes(x = mean_competition, y = coefficients))
p <- p + geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, size = 0.2)
```

```
p <- p + geom_point()
p <- p + theme_bw(base_size = 10)
p <- p + scale_x_continuous(breaks = c(3,4,5,6,7,8,9))
p <- p + ylab("Selection Gradient")
p <- p + ggtitle("Activity")
p <- p + xlab("Juvenile Competition")
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_text(size = 10),
  panel.border = element_rect(linetype = "solid", colour = "black"),
  plot.title = element_text(size = 10))
p <- p + geom_text(data = filter(sg.comp, variable == "Activity"),
  aes(x = 6, y = 2.2, label = paste("r = ", print, sep = '')), size = 2.5)
  pdf(file = "test.pdf", width = 2.17, height = 2.03)
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```

Docility and Competition

```
p <- ggplot(data = filter(sel_grads_blup_competition, variable == "Docility"),
  aes(x = mean_competition, y = coefficients))
p <- p + geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, size = 0.2)
p <- p + geom_point()
p <- p + theme_bw(base_size = 10)
p <- p + scale_x_continuous(breaks = c(3,4,5,6,7,8,9))
p <- p + ylab("Selection Gradient")
p <- p + ggtitle("Docility")
p <- p + xlab("Juvenile Competition")
p <- p + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), panel.background = element_blank(),
  strip.background = element_blank(), strip.text = element_text(size = 10),
  panel.border = element_rect(linetype = "solid", colour = "black"),
  plot.title = element_text(size = 10))
p <- p + geom_text(data = filter(sg.comp, variable == "Docility"),
  aes(x = 6, y = 1, label = paste("r = ", print, sep = '')), size = 2.5)
p + theme(plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "cm"))
```



Effect of competition on linear selection

```
load("data/analyses_data/fit_raneff_data.RData")
library(dplyr)
fit_raneff_data <- tbl_df(fit_raneff_data)
library(lme4)
fit_raneff_data$oID <- 1:nrow(fit_raneff_data)

ars_linear_model  <- glmer(ars_all ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s + (1 | grid_year) +
  (1|ID) + (1|oID), data = filter(fit_raneff_data, type == "blup"),
  family = poisson, control=glmerControl(optimizer="bobyqa"))
kpd_linear_model  <- glmer(kprod ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s + (1 | grid_year) +
  (1|ID) + (1|oID), data = filter(fit_raneff_data, type == "blup"),
  family = poisson, control=glmerControl(optimizer="bobyqa"))
ows_linear_model  <- glmer(prop ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s + (1 | grid_year) +
  (1|ID) + (1|oID), data = filter(fit_raneff_data, type == "blup"),
  weights = kprod, family = binomial, control=glmerControl(optimizer="bobyqa"))

save(ars_linear_model, kpd_linear_model, ows_linear_model,
  file = "data/analyses_data/ars_linear_blup_models.RData")
```

Models with observation level random effect.

```
load("data/analyses_data/fit_raneff_data.RData")
library(dplyr)
fit_raneff_data <- tbl_df(fit_raneff_data)
library(lme4)

fit_raneff_data$oID <- 1:nrow(fit_raneff_data)

ars_linear_model_  <- glmer(ars_all ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s + (1 | oID) +
  (1 | grid_year) + (1|ID), data = filter(fit_raneff_data, type == "blup"),
  family = poisson, control=glmerControl(optimizer="bobyqa"))
kpd_linear_model_  <- glmer(kprod   ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s + (1 | oID) +
  (1 | grid_year) + (1|ID), data = filter(fit_raneff_data, type == "blup"),
  family = poisson, control=glmerControl(optimizer="bobyqa"))
ows_linear_model_  <- glmer(prop    ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s + (1 | oID) +
  (1 | grid_year) + (1|ID), data = filter(fit_raneff_data, type == "blup"),
  weights = kprod, family = binomial, control=glmerControl(optimizer="bobyqa"))

save(ars_linear_model, kpd_linear_model, ows_linear_model,
  file = "data/analyses_data/ars_linear_blup_models.RData")

load("data/analyses_data/ars_linear_blup_models.RData")
library(lme4)
```

```
getLmerParams <- function(x){
  coefs <- summary(x)$coefficients
  coef.table <- data.frame(format(coefs[ ,1:3], digits = 1, nsmall = 2,
    scientific = FALSE))
  coef.table$pval[coefs[ ,4] > 0.001] <- format(coefs[coefs[ ,4] > 0.001, 4],
    digits = 1, nsmall = 2)
  coef.table$pval[coefs[ ,4] < 0.001] <- "< 0.001"
  coef.table$coefs <- paste(coef.table$Estimate, " ±", coef.table$Std..Error,
    sep = '')
  return(coef.table)
}

ars_linear_blup_results <- getLmerParams(ars_linear_model)
row.names(ars_linear_blup_results) <- c("Intercept", "Aggression",
  "Competition", "Activity", "Docility", "Aggression x Competition",
  "Activity x Competition", "Docility x Competition")
names(ars_linear_blup_results) <- c("Estimate", "SE", "Z", "P", "Est ± se")
```

Interaction between competition and linear selection results

```
pandoc.table(ars_linear_blup_results[c(1,3,2,4:8), c(5,4,3)],
  justify = "right", split.tables = 160)
```

	Est ± se	P	Z
Intercept	-1.10 ± 0.22	< 0.001	-5.09
Competition	-3.62 ± 0.64	< 0.001	-5.64
Aggression	0.74 ± 0.21	< 0.001	3.49
Activity	-0.32 ± 0.20	0.11	-1.61
Docility	-0.27 ± 0.16	0.10	-1.67
Aggression x Competition	2.62 ± 0.64	< 0.001	4.07
Activity x Competition	-1.36 ± 0.61	0.03	-2.21
Docility x Competition	-0.77 ± 0.49	0.11	-1.58

Competition and nonlinear selection

```
load("data/analyses_data/fit_raneff_data.RData")
library(lme4)
fit_blups_data <- filter(fit_raneff_data, type == "blup")
fit_blups_data$oID <- 1:nrow(fit_blups_data)

ars_nl_model <- glmer(ars_all ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s +
  aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
  I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
```

```
  (1 | grid_year) + (1|ID) + (1|oID),
  data = fit_blups_data, family = poisson,
  control=glmerControl(optimizer="bobyqa"))

ars_nl_model <- glmer(ars_all ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s +
  aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
  I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
  (1 | grid_year) + (1|ID),
  data = fit_blups_data, family = poisson,
  control=glmerControl(optimizer="bobyqa"))

kpd_nl_model <- glmer(kprod ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s +
  aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
  I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
  (1 | grid_year) + (1|ID) + (1|oID),
  data = fit_blups_data, family = poisson,
  control=glmerControl(optimizer="bobyqa"))

ows_nl_model <- glmer(prop ~ aggression_s*competition_s +
  activity_s*competition_s + docility_s*competition_s +
  aggression_s*activity_s*competition_s + I(aggression_s^2)*competition_s +
  I(activity_s^2)*competition_s + I(docility_s^2)*competition_s +
  (1 | grid_year) + (1|ID) + (1|oID),
  data = fit_blups_data, weights = kprod,
  family = binomial,
  control=glmerControl(optimizer="bobyqa"))

save(ars_nl_model,kpd_nl_model, ows_nl_model,
  file = "data/analyses_data/nl.blup_models.RData")

fit_raneff_data %>%
  ungroup() %>%
  summarise(
    mean_ars = mean(ars_all, na.rm = TRUE),
    var_ars = var(ars_all, na.rm = TRUE),
    mean_kpd = mean(kprod, na.rm = TRUE),
    var_kpd = var(kprod, na.rm = TRUE)
    )

## Source: local data frame [1 x 4]
##
##   mean_ars var_ars mean_kpd var_kpd
## 1   0.8784   1.255    3.902   4.372
```

Format results of nonlinear selection for table

```
load("data/analyses_data/nl.blup_models.RData")

coef_p_ars <- getLmerParams(ars_nl_model)
```

```
coef_p_ows <- getLmerParams(ows_nl_model)
coef_p_kpd <- getLmerParams(kpd_nl_model)

term_names <- c("Intercept", "Aggression",
 "Competition", "Activity", "Docility", "Aggression^2", "Activity^2",
 "Docility^2", "Aggression x Competition", "Activity x Competition",
 "Docility x Competition", "Aggression x Activity",
 "Aggression^2 x Competition", "Activity^2 x Competition",
 "Docility^2 x Competition", "Agg. x Act. x Competition"
 )

row.names(coef_p_ars) <- term_names
row.names(coef_p_ows) <- term_names
row.names(coef_p_kpd) <- term_names

names(coef_p_ars) <- c("Estimate", "SE", "Z", "P", "Est ± se")
names(coef_p_ows) <- c("Estimate", "SE", "Z", "P", "Est ± se")
names(coef_p_kpd) <- c("Estimate", "SE", "Z", "P", "Est ± se")
```

Nonlinear results

The effect of competition on linear and nonlinear selection on female behavioral traits for annual reproductive success. Posterior modes are given with highest posterior density intervals in parentheses.

ARS

```
pandoc.table(coef_p_ars[c(1,3,2,4:16), c(5,3,4)],
 split.tables = 160)
```

	Est ± se	Z	P
Intercept	-0.68 ± 0.31	-2.23	0.025
Competition	-2.24 ± 0.91	-2.46	0.014
Aggression	0.96 ± 0.31	3.06	0.002
Activity	-0.66 ± 0.28	-2.35	0.019
Docility	-0.51 ± 0.23	-2.18	0.029
Aggression^2	-0.65 ± 0.32	-2.07	0.039
Activity^2	-0.30 ± 0.26	-1.16	0.247
Docility^2	-0.18 ± 0.12	-1.48	0.139
Aggression x Competition	3.40 ± 0.92	3.68	< 0.001
Activity x Competition	-2.45 ± 0.84	-2.90	0.004
Docility x Competition	-1.41 ± 0.69	-2.04	0.041
Aggression x Activity	1.01 ± 0.43	2.35	0.019
Aggression^2 x Competition	-1.86 ± 0.96	-1.93	0.053

	Est ± se	Z	P
Activity^2 x Competition	-0.81 ± 0.77	-1.05	0.295
Docility^2 x Competition	-0.63 ± 0.39	-1.61	0.107
Agg. x Act. x Competition	2.62 ± 1.32	1.99	0.046

OWS

```r
pandoc.table(coef_p_ows[c(1,3,2,4:16), c(5,3,4)],
 split.tables = 160)
```

	Est ± se	Z	P
Intercept	-1.92 ± 0.39	-4.87	< 0.001
Competition	-2.48 ± 1.18	-2.11	0.035
Aggression	1.33 ± 0.42	3.17	0.001
Activity	-0.82 ± 0.38	-2.14	0.032
Docility	-0.49 ± 0.29	-1.70	0.088
Aggression^2	-1.02 ± 0.43	-2.36	0.018
Activity^2	-0.37 ± 0.36	-1.03	0.304
Docility^2	-0.18 ± 0.15	-1.17	0.242
Aggression x Competition	4.75 ± 1.28	3.70	< 0.001
Activity x Competition	-2.89 ± 1.20	-2.40	0.016
Docility x Competition	-0.78 ± 0.90	-0.86	0.388
Aggression x Activity	1.40 ± 0.60	2.35	0.019
Aggression^2 x Competition	-3.11 ± 1.36	-2.29	0.022
Activity^2 x Competition	-1.17 ± 1.10	-1.06	0.291
Docility^2 x Competition	-0.50 ± 0.50	-0.98	0.325
Agg. x Act. x Competition	3.95 ± 1.86	2.13	0.034

Fecundity

```r
pandoc.table(coef_p_kpd[c(1,3,2,4:16), c(5,3,4)],
 split.tables = 160)
```

	Est ± se	Z	P
Intercept	1.362 ± 0.091	14.937	< 0.001
Competition	-0.017 ± 0.161	-0.106	0.92

	Est ± se	Z	P
Aggregation	-0.058 ± 0.049	-1.176	0.24
Activity	0.030 ± 0.050	0.594	0.55
Docility	0.030 ± 0.041	0.724	0.47
Aggression^2	0.036 ± 0.053	0.672	0.50
Activity^2	-0.064 ± 0.051	-1.248	0.21
Docility^2	0.005 ± 0.026	0.209	0.83
Aggression x Competition	-0.067 ± 0.117	-0.579	0.56
Activity x Competition	0.008 ± 0.088	0.086	0.93
Docility x Competition	-0.116 ± 0.087	-1.325	0.19
Aggression x Activity	0.024 ± 0.075	0.319	0.75
Aggression^2 x Competition	0.153 ± 0.149	1.029	0.30
Activity^2 x Competition	0.079 ± 0.145	0.545	0.59
Docility^2 x Competition	-0.067 ± 0.055	-1.218	0.22
Agg. x Act. x Competition	-0.252 ± 0.240	-1.051	0.29

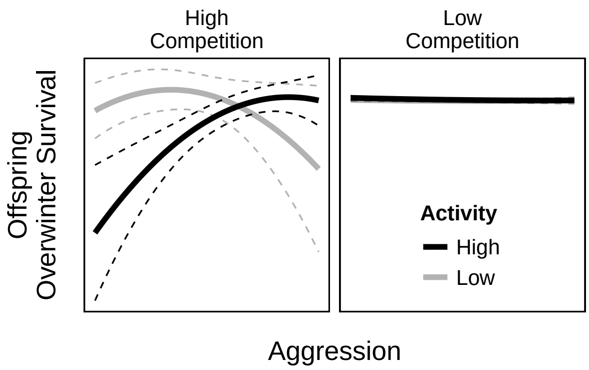
Plot of quadratic interaction

```
library(effects)

## Loading required package: colorspace
##
## Attaching package: 'effects'
##
## The following object is masked from 'package:car':
##
##     Prestige

library(ggplot2)

g.ows <- glm(prop ~ aggression_s * competition_s + activity_s *
    competition_s + docility_s * competition_s + aggression_s:activity_s *
    competition_s + I(aggression_s^2) * competition_s, data = filter(fit_raneff_data,
    type == "blup"), weights = kprod, family = binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

e.ows <- effect(term = "aggression_s:competition_s:activity_s",
    mod = g.ows)

e <- Effect(c("aggression_s", "activity_s", "competition_s"),
    g.ows, xlevels = list(aggression_s = 100, activity_s = 2,
```

```
        competition_s = 2))

s <- summary(e, type = "link")
se <- s$effect
su <- s$upper
sl <- s$lower
d <- as.data.frame(se)
du <- as.data.frame(su)
dl <- as.data.frame(sl)
names(d) <- c("la.lc", "ha.lc", "la.hc", "ha.hc")
names(du) <- c("la.lc", "ha.lc", "la.hc", "ha.hc")
names(dl) <- c("la.lc", "ha.lc", "la.hc", "ha.hc")

plot_d <- data.frame(Aggression = as.numeric(rep(row.names(s$effect),
    12)), OWS = c(d$la.lc, d$ha.lc, d$la.hc, d$ha.hc, du$la.lc,
    du$ha.lc, du$la.hc, du$ha.hc, dl$la.lc, dl$ha.lc, dl$la.hc,
    dl$ha.hc), Competition = rep(rep(c("Low\nCompetition", "High\nCompetition"),
    each = 200), 3), Activity = rep(rep(c("Low", "High"), each = 100),
    6), type = rep(c("main", "upper", "lower"), each = 400))
plot_d$env <- paste(plot_d$Competition, plot_d$Activity, sep = ".")

quad_plot <- ggplot(plot_d, aes(x = Aggression, y = OWS)) + geom_line(aes(alpha = Activity,
    linetype = type, size = type)) + facet_wrap(~Competition) +
    scale_alpha_discrete(range = c(1, 0.3)) + scale_linetype_manual(values = c(2,
    1, 2)) + scale_size_manual(values = c(0.3, 1, 0.3)) + ylab("Offspring\nOverwinter Survival") +
    xlab("Aggression") + theme_bw(base_size = 10) + theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.background = element_blank(),
    panel.border = element_rect(linetype = "solid", colour = "black"),
    axis.ticks = element_blank(), axis.text = element_text(size = 10),
    legend.key = element_blank(), strip.background = element_blank()) +
    guides(linetype = FALSE, size = FALSE, alpha = guide_legend(override.aes = list(size = 1))) +
    theme(legend.position = c(0.75, 0.25), legend.background = element_blank(),
        legend.key.size = unit(0.4, "cm")) + theme(plot.margin = unit(c(0.1,
    0.1, 0.1, 0.1), "cm")) + theme(axis.text.x = element_blank(),
    axis.text.y = element_blank())

pdf("figure/05_quad_print.pdf", width = 3.14, height = 2)
quad_plot
dev.off()

## pdf
##   2

quad_plot
```

Tile plot of interaction

```
n = 12
e <- Effect(c("aggression_s", "activity_s", "competition_s"),
    g.ows, xlevels = list(aggression_s = n, activity_s = n, competition_s = 2))
```




```
s <- summary(e, type = "response")
de <- as.data.frame(s$effect)
de_lc <- de[, 1:n]
de_hc <- de[, (n + 1):(2 * n)]

fix.names <- function(x) {
    a <- unlist(lapply(strsplit(names(x), split = "\\."), "[[",
        1))
    b <- unlist(lapply(strsplit(names(x), split = "\\."), "[[",
        2))
    out <- paste(a, b, sep = ".")
    return(out)
}

d_hc <- data.frame(Aggression = rep(row.names(de_hc), n), Activity = rep(fix.names(de_hc),
    each = n), OWS = as.vector(as.matrix(de_hc)))
d_hc$Aggression <- as.numeric(as.character(d_hc$Aggression))
d_hc$Activity <- as.numeric(as.character(d_hc$Activity))

d_lc <- data.frame(Aggression = rep(row.names(de_lc), n), Activity = rep(fix.names(de_lc),
    each = n), OWS = as.vector(as.matrix(de_lc)))
d_lc$Aggression <- as.numeric(as.character(d_lc$Aggression))
d_lc$Activity <- as.numeric(as.character(d_lc$Activity))

tile_plot <- ggplot(d_lc, aes(x = Aggression, y = Activity, z = OWS)) +
    geom_tile(aes(alpha = OWS), fill = "black", size = 0) + scale_alpha_continuous(range = c(1,
    0)) + ylab("Activity") + xlab("Aggression") + theme_bw(base_size = 10) +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), panel.border = element_rect(linetype = "solid",
            colour = "black"), axis.ticks = element_blank(),
        legend.key = element_blank(), strip.background = element_blank()) +
    guides(linetype = FALSE, size = FALSE, alpha = FALSE) + theme(plot.margin = unit(c(0.1,
    0.1, 0.1, 0.1), "cm"))

pdf("figure/05_tile_print.pdf", width = 3.14, height = 3.14)
tile_plot
dev.off()
```

```
## pdf
##   2
```

```
tile_plot
```