Contents

Random Effects, BLUPs & Repeatability

```
library(MASS) # MASS clashes with dplyr... so always load first
library(pander) # pander clashes with dplyr... so always load first
set.alignment('right', row.names = 'left')
library(dplyr)
library(MCMCglmm)
```

```
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 1: 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)

# 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 2: 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 3: 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)
```

```
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 4: Aggression repeatability using all trials

Aggression Repeatability Across Years

```
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 5: 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,
                          nitt = nitt,
                          verbose = FALSE
                          )
print(paste("Approx. model run time: ", format(Sys.time() - time_start)))
```

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

```
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 6: Aggression repeatability within years

Activity

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

```
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),
                    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"
```

```
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
```

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

Activity Repeatability

```
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 7: 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"
```

```
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 8: Activity repeatability across years

Activity Repeatabiltiy Within Years

```
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 9: 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))
  )
```

```
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 10: Summary of repeatabilities (continued below)

Run lmer models

```
library(lme4)
```

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

Docility

```
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

```
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

```
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)
  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

```
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.

```
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)
```

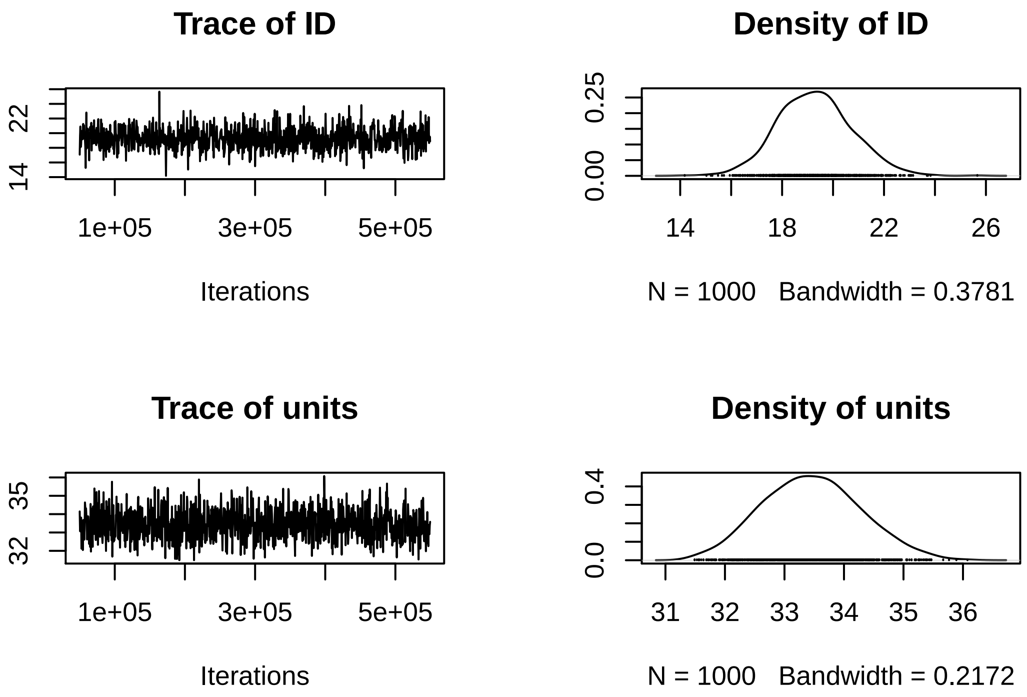
```
##
##       Stationarity start     p-value
##       test         iteration
## ID    passed       1         0.521
## units passed       1         0.706
##
##       Halfwidth Mean Halfwidth
##       test
## ID    passed    19.3 0.0897
## units passed    33.5 0.0506
```

```
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
```



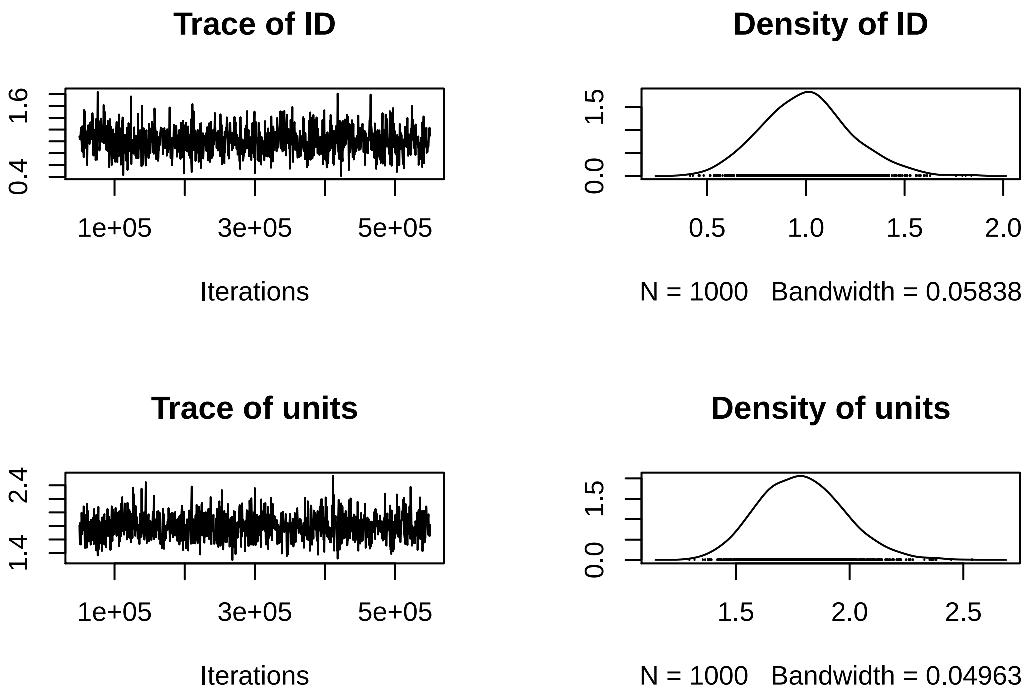
```
geweke.diag(agg_mcmc_model$VCV)
```

```
##
## 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
```

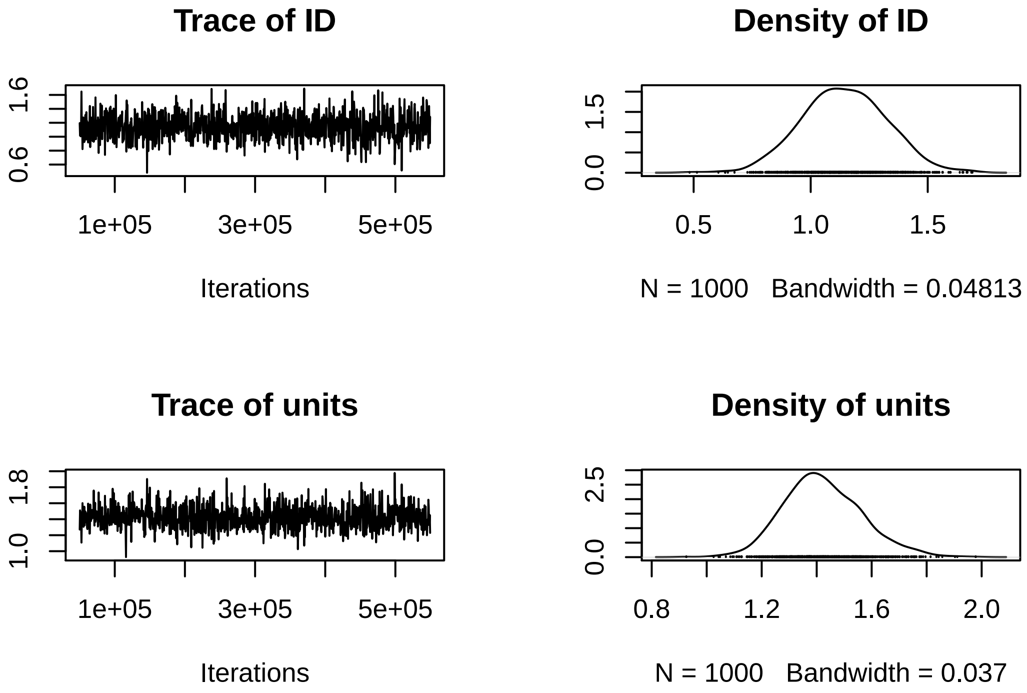
```
geweke.diag(act_mcmc_model$VCV)
```

```
##
## 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)
```


Trace of ID
1e+05
3e+05
5e+05
Iterations
Density of ID
N = 1000   Bandwidth = 0.04813
Trace of units
1e+05
3e+05
5e+05
Iterations
Density of units
N = 1000   Bandwidth = 0.037