Chemical signatures in the preen oil of Pied Flycatchers: testing reproducibility and exploring ontogeny

Laurence Jeanjean, Barbara A. Caspers, Tim Schmoll & Marc Gilles

2024-10-07

Contents

- Context

The present document is an appendix of a paper on the preen oil composition in pied flycatchers (reproducibility and ontogeny). It contains the script for all R-based analyses from this paper. In the first part of this document, we prepare the chemical and meta data for analysis. In a second part, we present the code and results of our pre-registered analysis, replication of an original study by Gilles et al. (2024). In a last part you will find the code and results for our exploratory analyses on the same data-set.

- Pre-registered analysis: https://osf.io/tbcug
- Original paper: Gilles M, Fokkema RW, Krosten P, Caspers BA, Schmoll T. Preen oil composition of Pied Flycatchers is similar between partners but differs between sexes and breeding stages. Ibis. 2024.

1 Data

1.1 Import library and data

Required library

```
library(tidyverse) # For data transformation and plots
library(readxl) # To upload excel files
library(lme4) # To fit the mixed model
library(lmerTest) # To obtain P-values of the mixed models
library(broom.mixed) # To obtain $\beta$ estimates and their confidence intervals of the fixed effects
library(partR2) # To obtain marginal R2
library(rptR) # To obtain repeatability of random effects
library(performance) # For model diagnostics
#library(qqplotr)  # For the performance package to fully function
library(see) # For the performance package to fully function
library(patchwork)  # For the performance package to fully function
library(effsize)  # For effect sizes
library(vegan) # For Bray Curtis matrices in the spatial analysis
library(glmmTMB)  # For Pairwise distance analyses
library(GCalignR) # For the chemical alignment
```

Import data

```
setwd("~/your_own_file")
Metadata <- read_excel("metadata.xlsx")
ChemdataRaw <- read.csv("chemdata.csv", header = F, check.names = F)
MetadataP1 <- read_excel("metadata_orig.xlsx")
eff_sizes <- read_excel("effect_sizes.xlsx")
GPSNetboxes <- read.csv("gpsdata.csv")
```

1.2 Pre-selection of samples for analysis

We discard N=23 samples for which the chromatogram seemed to carry too much noise or no preen oil substances, N=2 samples for which information on breeding stage was missing, and N=6 samples from individuals that were sampled twice during nestling-rearing (in that case we kept the second sample, the first one being too close to the hatching date). We are left with a total of 218 samples.

```
Metadata <- Metadata %>%
  filter(GCsuccess == 1,  # Remove samples with too much noise or no preen oil
         Breeding_Stage != "NA", # Remove samples with no information on breeding stage
         CapturedTwiceNR == 0) %>% # Remove samples from individuals sampled twice during nestling-rear
  select(-c(GCsuccess, CapturedTwiceNR, captureevent, wing, mass, drost, GCbatch)) # Remove variables t
```

1.3 Preparation of the Chemical data

1.3.1 Alignment of the chemical data with GCalignR

```
#Creating a file in the right format for GCalignR
alignd ata <- ChemdataRaw[,!(ChemdataRaw[3,]=="failed chromatogram")] # Removing failed samples
alignd ata <- alignd ata[,!(alignd ata[3,]=="no chromatogram")]
alignd ata <- alignd ata[,c(T,T,F,F)] # Removing 3rd and 4th colomn
sampleids <- alignd ata[1,] # create a dataframe with just the first row
sampleids <- sampleids[,c(T,F)] # remove empty cell between each sampleID
RTarea <- matrix(c("RT","area")) # prepare the row with RT/area
RTarea <- t(RTarea) # flip columns and rows
RTarea <- data.frame(RTarea)
alignd ata <- qpcR:::rbind.na(sampleids, RTarea, alignd ata) # bind sampleIDs, RTarea and the data, fill
ChemdataRaw <- alignd ata[-c(3,4),] # remove unnecessary rows
write.table(ChemdataRaw,"Alignment.txt",
            row.names = F, col.names = F, sep="\t", na = "", quote = F) # save as a .txt file
check_input("Alignment.txt") # check that the data format is good

#Choosing a reference sample
choose_optimal_reference(data = "Alignment.txt", rt_col_name = "RT")

#Grouping blanks and field controls in a separate file
blanks <- c("blank1108_1", "blank1108_2", "blank1208_1", "blank1208_5", "blank1308_1", "blank1308_2", "
            "L21", "L31", "L49", "L62", "L44", "L70", "L82", "L91", "L97", "L133", "L166", "L199", "L23

#Alignment
alignedData <- align_chromatograms(
  data = "Alignment.txt", # raw chromatographic data
  rt_col_name = "RT",
  reference = "L223", # obtained from the function 'choose_optimal_reference()'
  blanks = blanks, # delete substances detected in control samples
  delete_single_peak = TRUE,  # delete substances detected in one sample only
  remove_empty = TRUE,  # remove empty samples
  max_linear_shift = 0.03, # expected linear drift
  max_diff_peak2mean = 0.015, # allowed RT difference of a peak with the mean of the corresponding row
  min_diff_peak2peak = 0.035, # expected minimum RT difference among homologous substances
  permute = F,  # keep the order of samples constant between different alignments
  write_output = c("area"))

print(alignedData)
save(alignedData, file = "alignedData.RData")

# Diagnistics plots
gc_heatmap(alignedData,threshold = 0.02)
plot(alignedData,which_plot = "all")
```

If the chemical data has already been aligned once on the computer, the aligned data can directly be loaded (instead of running the alignment each time)

```
load("~/your_own_file/alignedData.RData")
```

Manual method (load the data from the text file created by GC-alignR)

```
Chemdata <- read.table("Alignment_area.txt",header = F)
samplenames <- Chemdata[1,]
```

```
rownames(Chemdata) <- as.matrix(Chemdata[,1])
Chemdata <- Chemdata[,-1]
colnames(Chemdata) <- as.matrix(Chemdata[1,])
Chemdata <- Chemdata[-1,]
Chemdata <- as.data.frame(t(Chemdata))
str(Chemdata)
chemdata.num <- data.frame(lapply(Chemdata, function(x) as.numeric(as.character(x))))
str(chemdata.num)
chemdata.num$sample <- t(samplenames)[-1,]
chemdata.num[names(chemdata.num)=='sample']
chemdata.num <- chemdata.num[,c(which(colnames(chemdata.num)=="sample"),
                                which(colnames(chemdata.num)!="sample"))]
rownames(chemdata.num) <- as.matrix(chemdata.num[,1])
chemdata.num <- chemdata.num[,-1]
Chemdata <- chemdata.num
noms_colonnes <- names(Chemdata)
noms_colonnes_sans_X <- sub("^X", "", noms_colonnes)
names(Chemdata) <- noms_colonnes_sans_X
```

After alignment, 8 samples appeared as clear outliers (4 samples from incubating females, 1 sample from an incubating male and 3 samples from nestlings). They are not included in the data subsets of the different analyses.

1.3.2 Transformation of the aligned data

Standardize the aligned data

```
Chemdata.norm <- norm_peaks(alignedData, conc_col_name = "area",rt_col_name = "RT",
out = "data.frame")
```

Log-transforme the aligned data

```
Chemdata <- log(Chemdata.norm + 1)
```

1.4 Preparation of the Metadata

1.4.1 Calculating alpha diversity and Volatility measures

Chemical richness: the number of substances in each sample

```
Richness <- specnumber(Chemdata)
```

Shannon Diversity of each sample

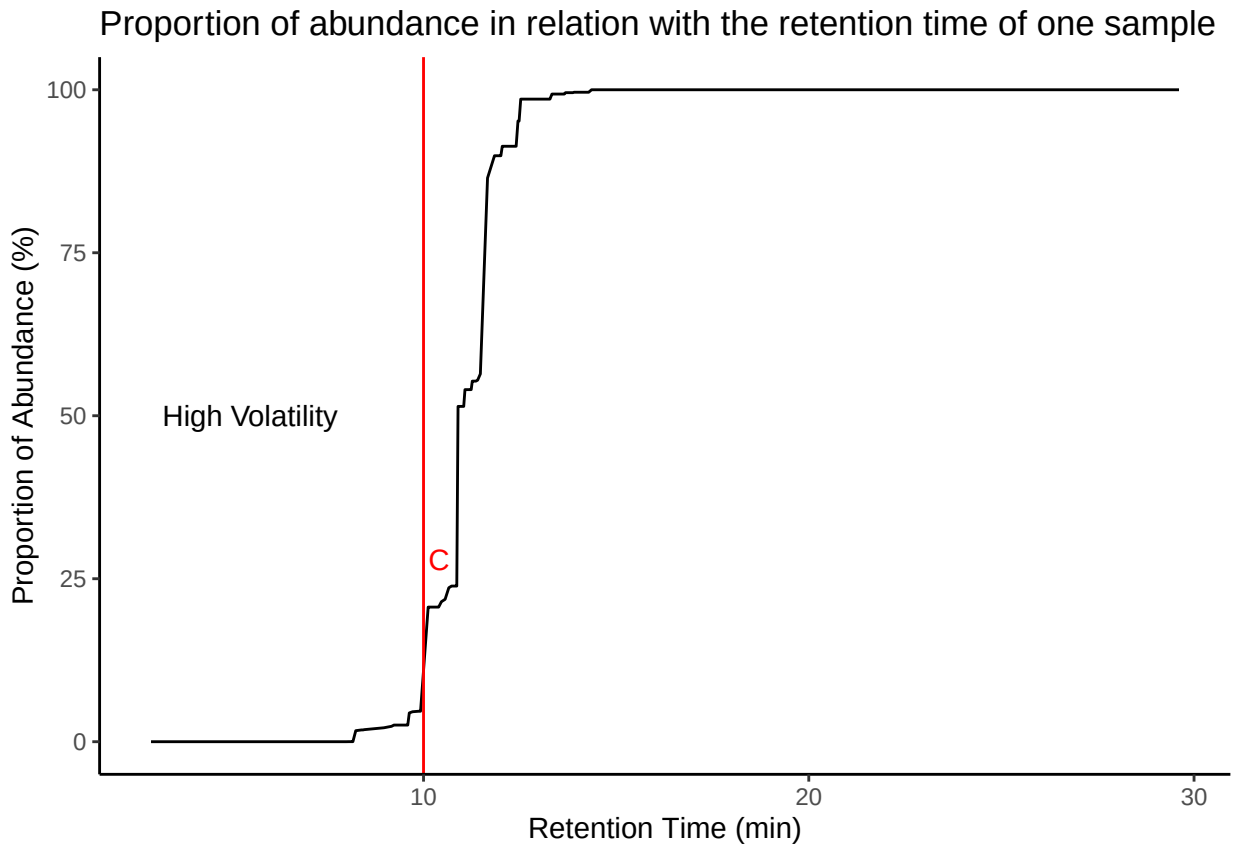
```
Shannon_Index <- diversity(Chemdata)
```

Volatility: i.e. the proportion of highly volatile compounds (total area under the chromatogram before peak C (retention time 10.12))

```
# We get the volatility on the chemical data non log-transformed (Chemdata.norm)
CumSum <- t(apply(Chemdata.norm, 1, cumsum))
volatility <- CumSum[,"9.922", drop = F]
colnames(volatility) <- "Volatility"
volatility <- as.data.frame(volatility)
```

Example of the cumulative abundance curve of one sample for visualisation

```
RT <- as.numeric(colnames(CumSum))
prop <- CumSum["L61",] # sample L61
data.plot <- data.frame(RT, prop)

theme_set(theme_classic())
ggplot(data.plot, aes(x=RT, y=prop))+
  geom_line()+
  geom_vline(xintercept=10, color="red")+
  xlab("Retention Time (min)")+
  ylab("Proportion of Abundance (%)") +
  annotate("text", x=10.4, y=28, label= "C", color="red")+
  annotate("text", x=5.5, y=50, label= "High Volatility")+
  ggtitle("Proportion of abundance in relation with the retention time of one sample")
```



Add the alpha-diversity and volatility measures to the Metadata

```
alpha_diversity <- data.frame(Richness = Richness, Shannon_Index = Shannon_Index)
Metadata <- cbind(Metadata, alpha_diversity)
Metadata <- cbind(Metadata, volatility)

# Transform the alpha-diversity and volatility measures as numeric variables
Metadata <- Metadata%>%
  mutate(Richness = as.numeric(as.character(Richness)),
         Shannon_Index = as.numeric(as.character(Shannon_Index)),
         Volatility = as.numeric(as.character(Volatility)))
```

How many substances:

```
ncol(Chemdata)
```

```
## [1] 88
```

```
mean(Metadata$Richness)
```

```
## [1] 24.45872
```

```
sd(Metadata$Richness)
```

```
## [1] 7.448789
```

On the 218 samples retained for alignment, we find 88 substances. On average, each sample has 24 substances (sd = 7).

```
# Create a Sample data set and a Sample variable in Chemdata
Sample <- rownames(Chemdata)
Chemdata <- rownames_to_column(Chemdata, var = "Sample")
```

1.5 Control for Concentration bias

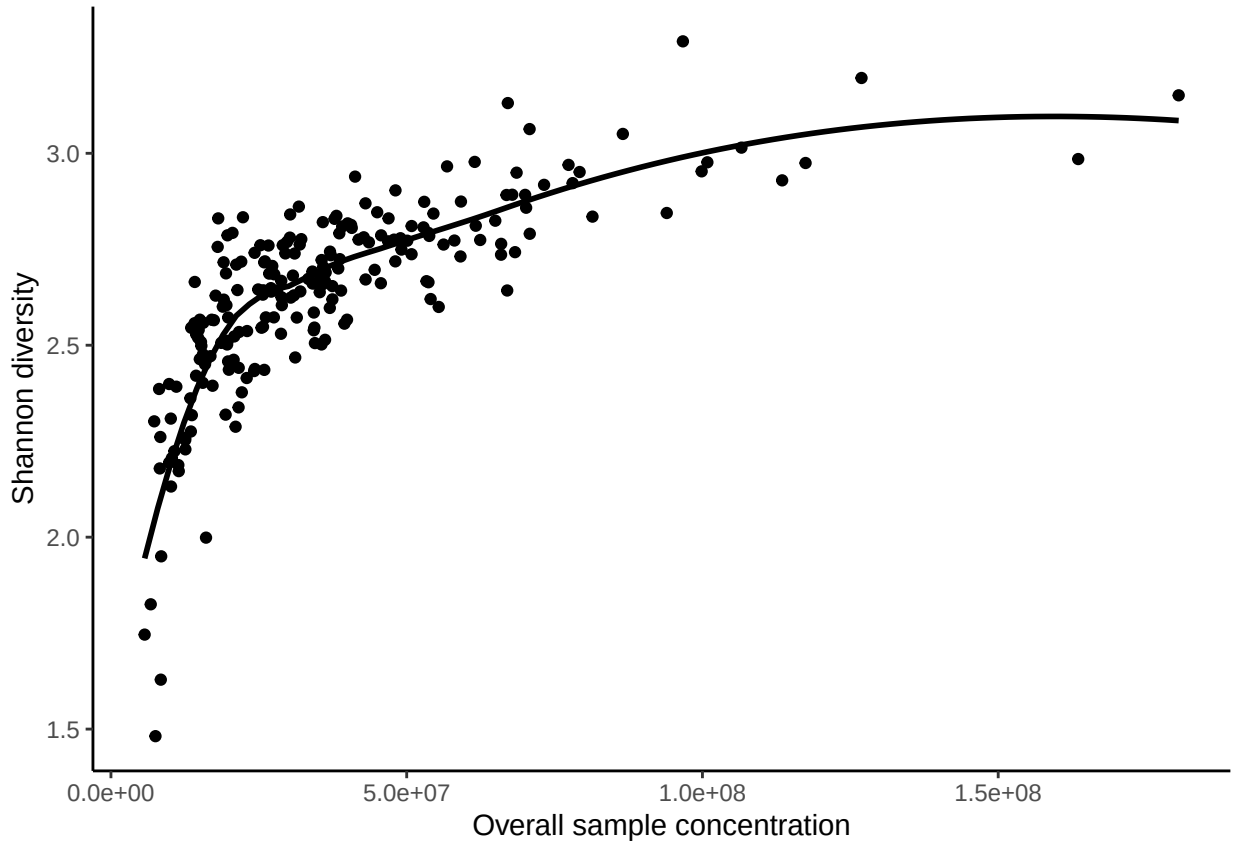
Plot the Shannon diversity against "concentration", i.e. the total area under the chromatogram divided by the number of substances (the mean area under a substance in each sample)

```
# We need to calculate the area under the entire chromatogram, therefore we use
#  the raw chemical data, before alignment.
TotArea <- ChemdataRaw[,!(ChemdataRaw[3,]=="failed chromatogram")]
TotArea[1, seq(2, ncol(TotArea), by = 2)] <- TotArea[1, seq(1, ncol(TotArea), by = 2)] # So that the ar
TotArea <- TotArea[,c(F,T,F,F)] # select the areas
TotArea <- TotArea[-2,]
TotArea <- t(TotArea)
TotArea <- as.data.frame(TotArea)
SampleTotArea <- TotArea[,1]
SampleTotArea <- as_tibble(SampleTotArea)
SampleTotArea <- SampleTotArea %>% rownames_to_column()
TotArea <- as_tibble(TotArea)
TotArea <- TotArea%>% # Calculate the total area under each chromatogram
  mutate_all(funs(as.integer(as.character(.))))%>%
```

```
  rowwise() %>%
  mutate(AreaTot= sum(c_across(), na.rm = T))%>%
  dplyr::select("AreaTot", everything())
TotArea <- TotArea%>%
  select(AreaTot)%>%
  rownames_to_column()
df_list <- list(SampleTotArea, TotArea)
TotArea <- df_list%>%
  reduce(full_join, by="rowname")%>%
  select(value, AreaTot)%>%
  rename(Sample = value)
df_list <- list(Metadata, TotArea)
plot.Concentration <- df_list%>% # Combine with the metadata
  reduce(full_join, by="Sample")%>%
  filter(Individual_ID != "NA")%>% # select only the samples that  get used during analyses
  mutate(Concentration = AreaTot/Richness) # calculate our proxy for the concentration of each sample
```

```
# plot
ggplot(plot.Concentration, aes(x=Concentration, y=Shannon_Index))+
  geom_point(shape=19)+
  geom_smooth(color="black", se=F)+
  ylab("Shannon diversity")+
  xlab("Overall sample concentration")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Unlike the original study, although there is a positive relationship between "concentration" and Shannon diversity, there is no clear concentration threshold under which the Shannon diversity drops. Therefore, we will not discard additional samples from our data set.

2 Pre-registered analysis

You will find here the NMDS and GLMM analysis that we replicated from Gilles et al. (2024). The PERMANOVA and PERMDISP analyses were conducted using PRIMER v7.0.21, and therefore are not available on this document.

We studied the effects of sex (fixed) and pair identity (random), as well as the effects of breeding stage (fixed) and individual identity (random) on the chemical richness, Shannon diversity and volatility of preen oil using linear mixed models (LMM) with Gaussian distributions, using the *lme4* package (Bates et al. 2014). We assessed the significance of fixed effects by checking whether the 95% confidence interval of the beta estimates contained 0 using the broom.mixed package (Bolker et al. 2022), and also checked P-values using the *lmerTest* package (Kuznetsova et al. 2017). The significance of random effects was evaluated by checking whether the 95% confidence intervals of the repeatability estimates contained 0, and by checking the P-value based on permutations, using the *rptr* package (Stoffel et al. 2017). In addition, we measured the variance explained (marginal R^2) by each fixed effect using the *partR2* package (Stoffel et al. 2021). We verified the assumptions for LMMs using the *performance* package (Lüdecke et al. 2021).

2.1 Sex and breeding pair effects during nestling-rearing:

Data = 46 breeding pairs (92 samples) sampled during the nestling-rearing period

```
# Subset of the Metadata for the sex and breeding pair analyses:
Pairs_Nrearing <-Metadata%>%   #92 samples
  filter(pair_brood==1)%>%
  select(-c(Outliers, F_Connected_to_Outlier, Partner_Connected_to_Outlier, f_sampled_twice, Families, I
```

2.1.1 NMDS plots

Here we create NMDS plots showing the similarity between our samples on a 2D scale. We use Bray-Curtis distances as our similarity measure.

Step 1: Building a Bray-Curtis matrix for our 92 samples

```
ChemdataSex <- Chemdata%>%
  filter(Sample %in% Pairs_Nrearing$Sample)%>%
  select(-Sample)# Select the subset of Chemdata from our 92 samples
bc <- metaMDS(ChemdataSex, distance = "bray") # Bray-Curtis matrix
```

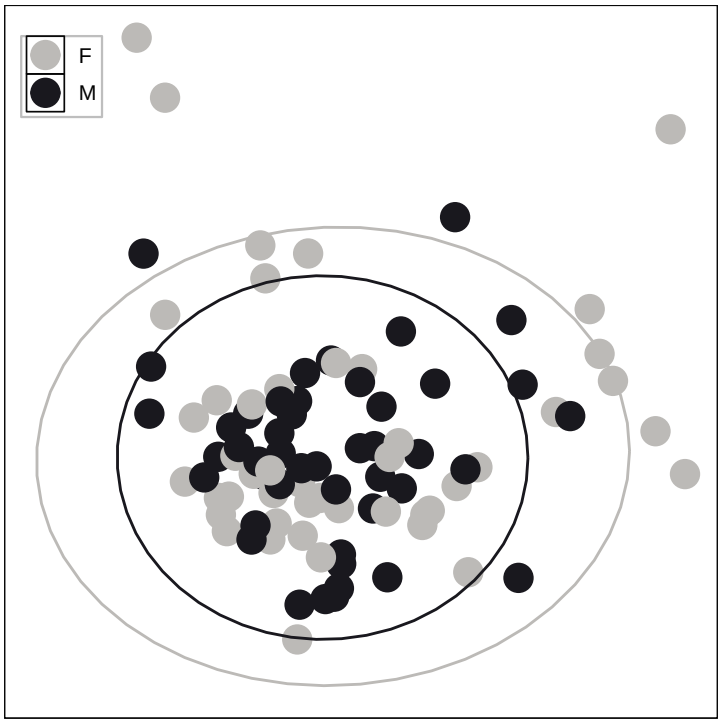
Step 2: Checking the stress (how good the distance between samples in actual multivariate distance is represented in two dimensions)

```
bc$stress
```

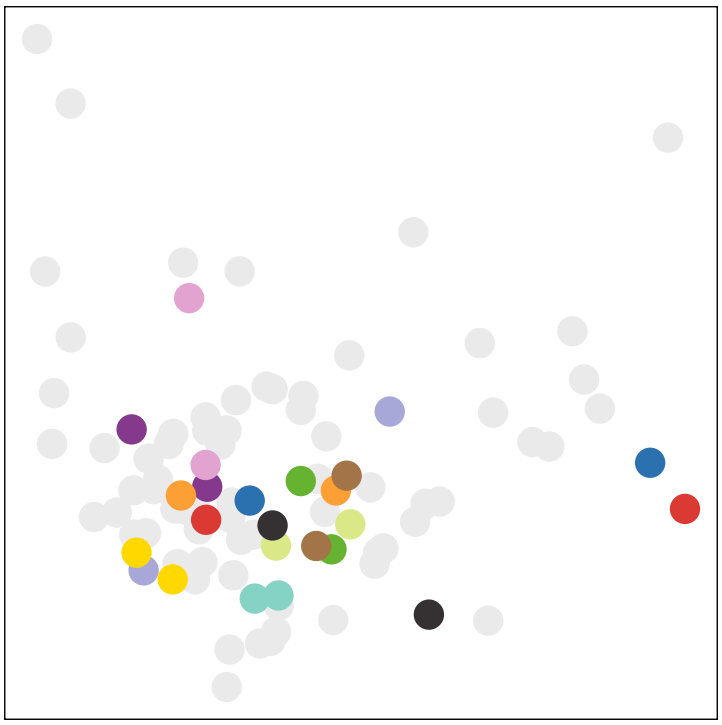
```
## [1] 0.07472752
```

Step 3: Plot the NMDS, here by sex

```
bc <- as.data.frame(bc[["points"]]) # Create a data file with the coordinates of each samples in the Br
# Add the coordinates of each sample in the Subset of the Metadata
Pairs_Nrearing$MDS1 <- bc$MDS1
Pairs_Nrearing$MDS2 <- bc$MDS2
# plot
ggplot(Pairs_Nrearing) +
  geom_point(aes(x=MDS1, y=MDS2, color = Sex, size = Sex, shape = Sex)) +
  stat_ellipse(aes(MDS1, MDS2, color = Sex), type = "t", level = 0.95) +
  scale_size_manual(values=c(5,5)) +
  scale_shape_manual(values=c(16,16)) +
  scale_color_manual(values = c("#BCBAB7","#19181E")) +
  theme_void() +
  theme(panel.background = element_rect(colour = "grey3", size = 0.3, fill = NA),
        aspect.ratio = 1,
        legend.title = element_blank(),
        legend.position = c(0.08, 0.9),
        legend.text = element_text(size=8),
        legend.background = element_rect(size = 0.4, linetype = "solid", color = "grey"),
        legend.key.size = unit(0.3, "cm"),
        legend.margin = margin(0,2,2,2),
        plot.margin=unit(c(1,1,1,1),"cm"))
```



We can see here that there seem to be no difference in position or dispersion between male and female samples, as confirmed by the PERMANOVA and PERMDISP analyses on PRIMER.



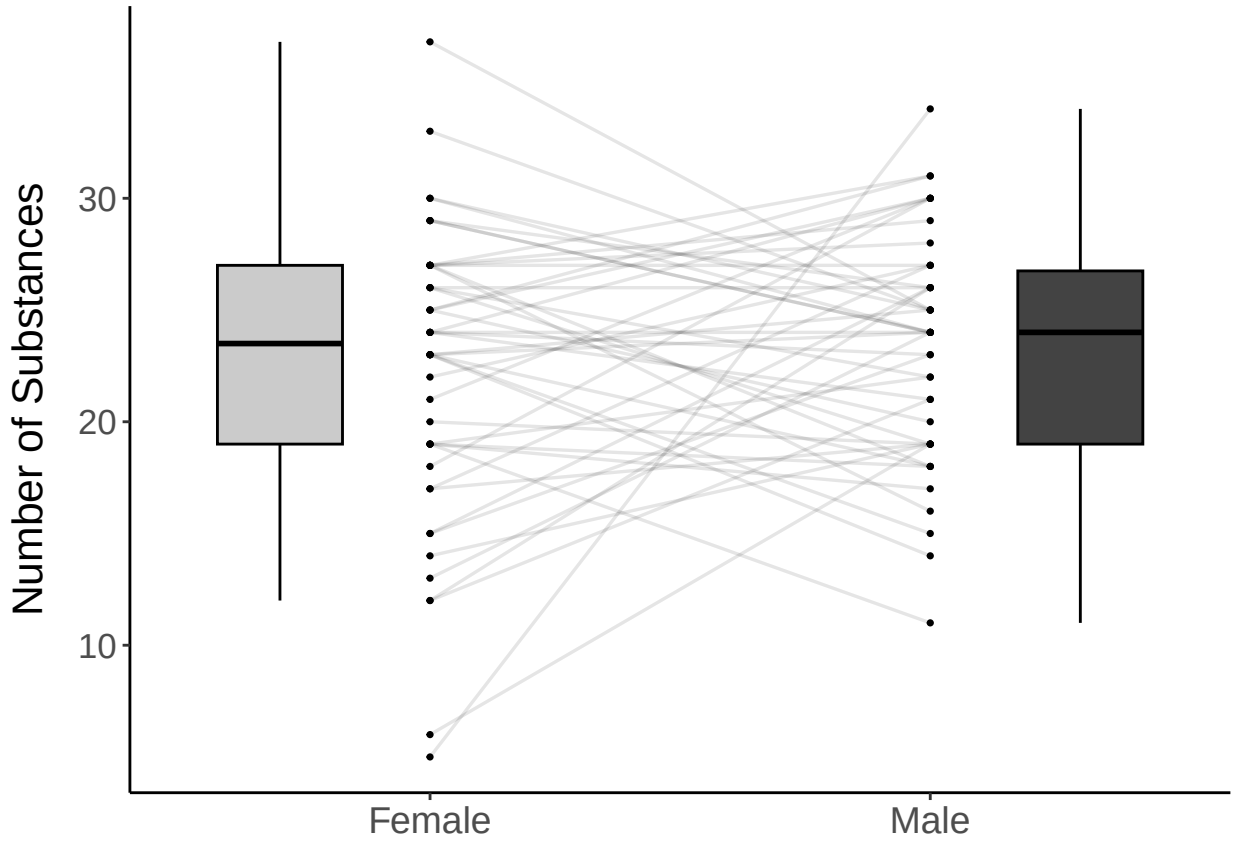
The PERMANOVA analysis also shows that breeding pairs are significantly more similar to each other than random. Therefore, we also create an NMDS plot by breeding pairs. To make the figure more easy to read, we choose randomly 12 breeding pairs that will be represented in the plot.

2.1.2 Richness (Number of substances recorded after alignment)

a. Visualisation of the sex difference (within pairs) in terms of chemical richness.

```
theme_set(theme_classic())
ggplot(Pairs_Nrearing, aes(x=Sex, y=Richness))+
  geom_boxplot(data = Pairs_Nrearing %>% filter(Sex=="F"),
               aes(x=Sex, y=Richness),
               position=position_nudge(x=-0.3), outlier.shape = NA, width = .25, lwd = 0.5,
               alpha = .8, colour="black", fill = "grey")+
  geom_boxplot(data = Pairs_Nrearing %>% filter(Sex=="M"), lwd = 0.5,
               aes(x=Sex, y=Richness),
               position=position_nudge(x=0.3), outlier.shape = NA, width = .25,
               alpha = .8, colour="black", fill = "grey8")+
  geom_line(aes(group=Nestbox_ID), alpha=0.1, size = 0.6, position=position_dodge(0), show.legend = FALS
  geom_point(size = 0.6)+
  scale_x_discrete(labels=c("Female","Male")) +
  ylab("Number of Substances")+
  theme(axis.title.y=element_text(size=16, margin = margin(t = 0, r = 10, b = 0, l = 0)),
        axis.text.y = element_text(size=13),
```

```
        axis.text.x = element_text(size=14),
        axis.title.x = element_blank())
```



In this boxplot, each point represents a sample, and each line connects the members of a breeding pair.

b. Analysis

- Fitting the model

```
LMM_Sex_R <- lme4::lmer(formula = "Richness ~ Sex + (1 | Nestbox_ID)",  data = Pairs_Nrearing)
summary(LMM_Sex_R)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Richness ~ Sex + (1 | Nestbox_ID)
##    Data: Pairs_Nrearing
##
## REML criterion at convergence: 581.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9701 -0.6018  0.1371  0.6521  2.4145
##
## Random effects:
```

```
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept)  2.123   1.457
##  Residual               32.441   5.696
## Number of obs: 92, groups:  Nestbox_ID, 46
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  22.3043     0.8668  25.731
## SexM          1.0870     1.1876   0.915
##
## Correlation of Fixed Effects:
##      (Intr)
## SexM -0.685
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Richness ~ Sex + (1 | Nestbox_ID)",  data = Pairs_Nrearing))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Richness ~ Sex + (1 | Nestbox_ID)"
##    Data: Pairs_Nrearing
##
## REML criterion at convergence: 581.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9701 -0.6018  0.1371  0.6521  2.4145
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept)  2.123   1.457
##  Residual               32.441   5.696
## Number of obs: 92, groups:  Nestbox_ID, 46
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  22.3043     0.8668 89.6618  25.731   <2e-16 ***
## SexM          1.0870     1.1876 45.0000   0.915    0.365
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## SexM -0.685
```

P-value = 0.365 –> non significant.

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_Sex_R, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term            estimate std.error stati~1 conf.~2 conf.~3
##   <chr>    <chr>      <chr>              <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>       (Intercept)        22.3      0.867  25.7     20.3    23.8
## 2 fixed    <NA>       SexM                1.09     1.19    0.915   -1.49    3.28
## 3 ran_pars Nestbox_ID sd__(Intercept)     1.46    NA      NA        0       3.73
## 4 ran_pars Residual   sd__Observation     5.70    NA      NA        4.55    6.61
## # ... with abbreviated variable names 1: statistic, 2: conf.low, 3: conf.high
```

β estimate of sexM effect: 1.09 –> males have on average 1.09 more substances than females in our samples. Confidence interval: [-1.60 ; 3.38] –> includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_Sex_R, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.0086 0        0.0771   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of sex: 0.0086 The sex effect only explains 0.86% of the variation of richness in adults during nestling-rearing.

- Finding the repeatability (random effect)

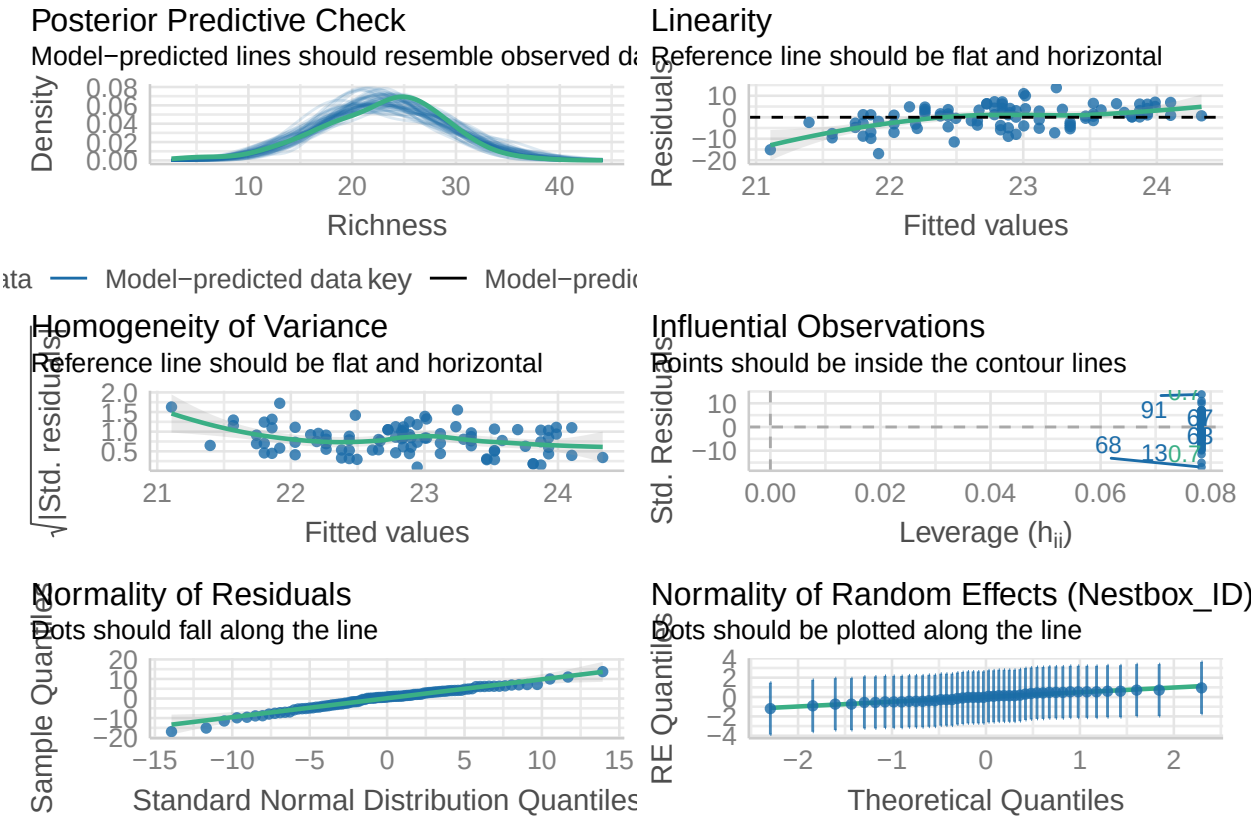
```
rpt(Richness ~ Sex + (1 | Nestbox_ID),
    grname = "Nestbox_ID",
    data = Pairs_Nrearing, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Nestbox_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Nestbox_ID
## R  = 0.061
## SE = 0.109
## CI = [0, 0.352]
## P  = 0.338 [LRT]
##      0.381 [Permutation]
```

Repeatability of the NestboxID effect: 0.061 –> 6.1% of the variation of richness between samples is due to the variation between pairs in our data. P(perm)= 0.342, non significant. The chemical richness is not repeatable between partners.

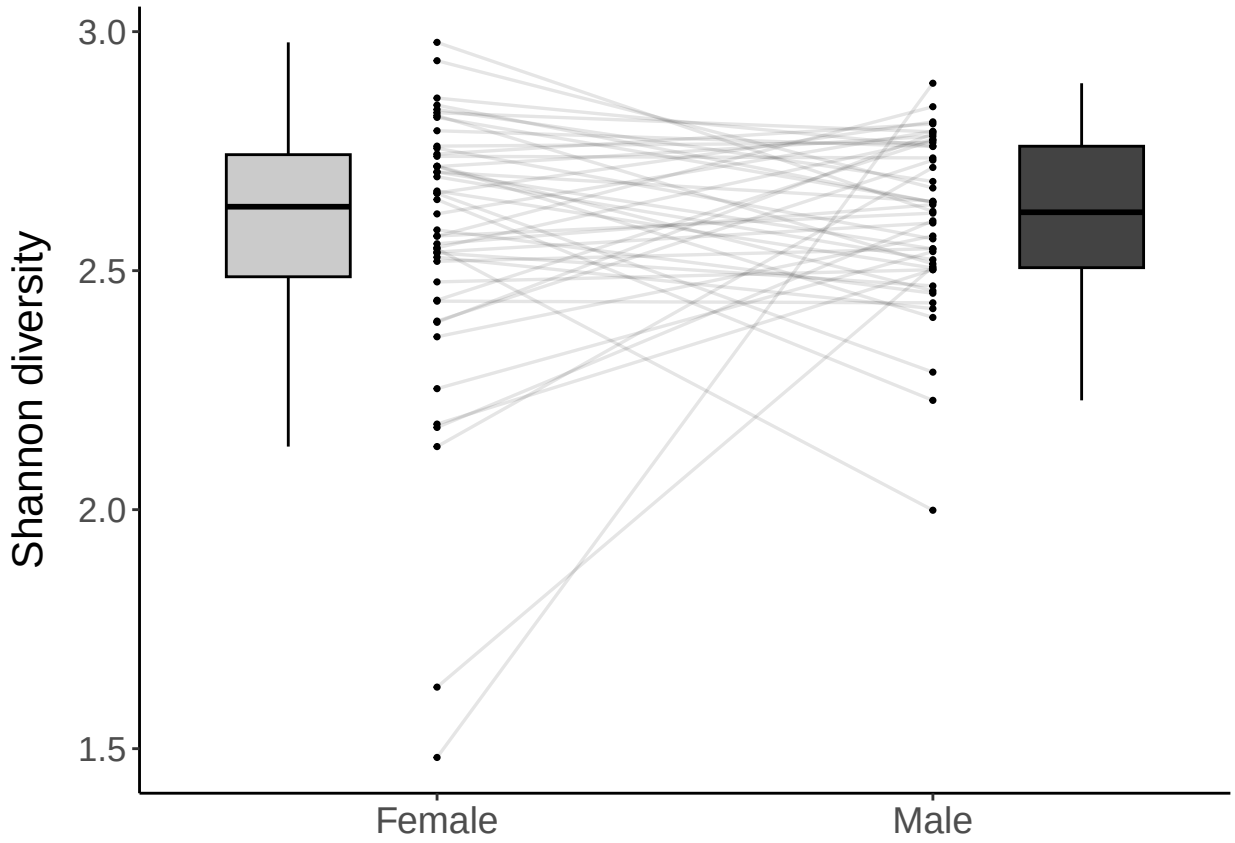
- Model diagnostic

```
check_model(LMM_Sex_R)
```



2.1.3 Diversity (Shannon Index)

a. Visualisation of the sex difference (within pairs) in terms of Shannon diversity.



b. Analysis

- Fitting the model

```
LMM_Sex_D <- lme4::lmer(formula = "Shannon_Index ~ Sex + (1 | Nestbox_ID)",  data = Pairs_Nrearing)
summary(LMM_Sex_D)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Shannon_Index ~ Sex + (1 | Nestbox_ID)
##    Data: Pairs_Nrearing
##
## REML criterion at convergence: 8.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4774 -0.4051  0.1037  0.6733  1.6916
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.00000  0.0000
##  Residual               0.05881  0.2425
## Number of obs: 92, groups:  Nestbox_ID, 46
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)  2.56738    0.03576  71.804
## SexM         0.03856    0.05057   0.763
##
## Correlation of Fixed Effects:
##      (Intr)
## SexM -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Shannon_Index ~ Sex + (1 | Nestbox_ID)",  data = Pairs_Nrearing))
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Shannon_Index ~ Sex + (1 | Nestbox_ID)"
##    Data: Pairs_Nrearing
##
## REML criterion at convergence: 8.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4774 -0.4051  0.1037  0.6733  1.6916
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.00000  0.0000
##  Residual               0.05881  0.2425
## Number of obs: 92, groups:  Nestbox_ID, 46
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  2.56738    0.03576 90.00000  71.804   <2e-16 ***
## SexM         0.03856    0.05057 90.00000   0.763    0.448
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## SexM -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

P-value = 0.448 –> non significant.

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_Sex_D, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term            estimate std.error stati~1 conf.~2 conf.~3
##   <chr>    <chr>      <chr>              <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>       (Intercept)       2.57      0.0358  71.8    2.50     2.64
## 2 fixed    <NA>       SexM              0.0386    0.0506   0.763 -0.0652   0.135
## 3 ran_pars Nestbox_ID sd__(Intercept)   0        NA       NA      0        0.142
## 4 ran_pars Residual   sd__Observation   0.243    NA       NA      0.188    0.269
## # ... with abbreviated variable names 1: statistic, 2: conf.low, 3: conf.high
```

β estimate of sexM effect: 0.038 –> males have an average greater diversity of 0.038 than females in our samples. Confidence interval: [-0.059 ; 0.14] –> includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_Sex_D, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.0064 0        0.0726   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of sex: 0.0064 So the sex only explain 0.64% of the variation of the diversity in our data.

- Finding the repeatability (random effect)

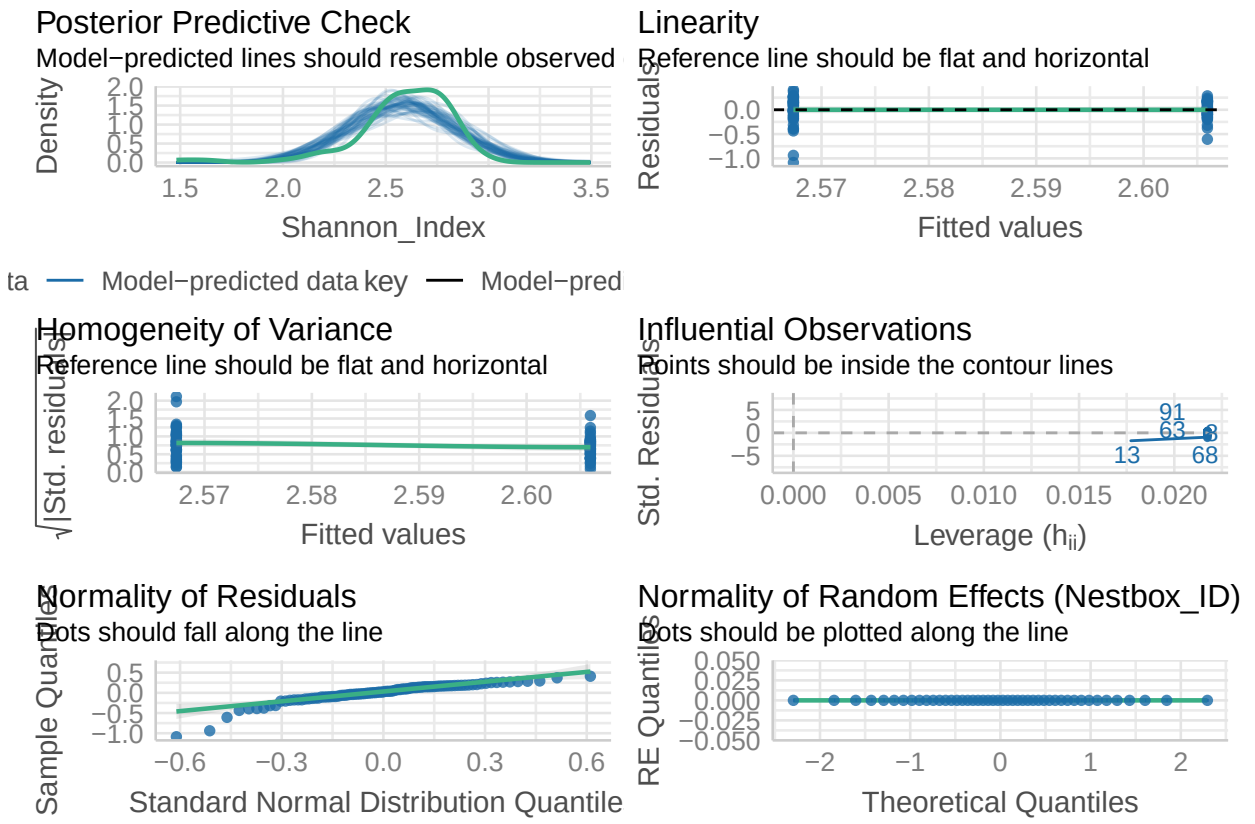
```
rpt(Shannon_Index ~ Sex + (1 | Nestbox_ID),
    grname = "Nestbox_ID",
    data = Pairs_Nrearing, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Nestbox_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Nestbox_ID
## R  = 0
## SE = 0.09
## CI = [0, 0.297]
## P  = 1 [LRT]
##      1 [Permutation]
```

Repeatability of the NestboxID effect: 0 –> 0% of the variation of diversity between samples is due to the variation between pairs in our data. The diversity is not repeatable between partners.

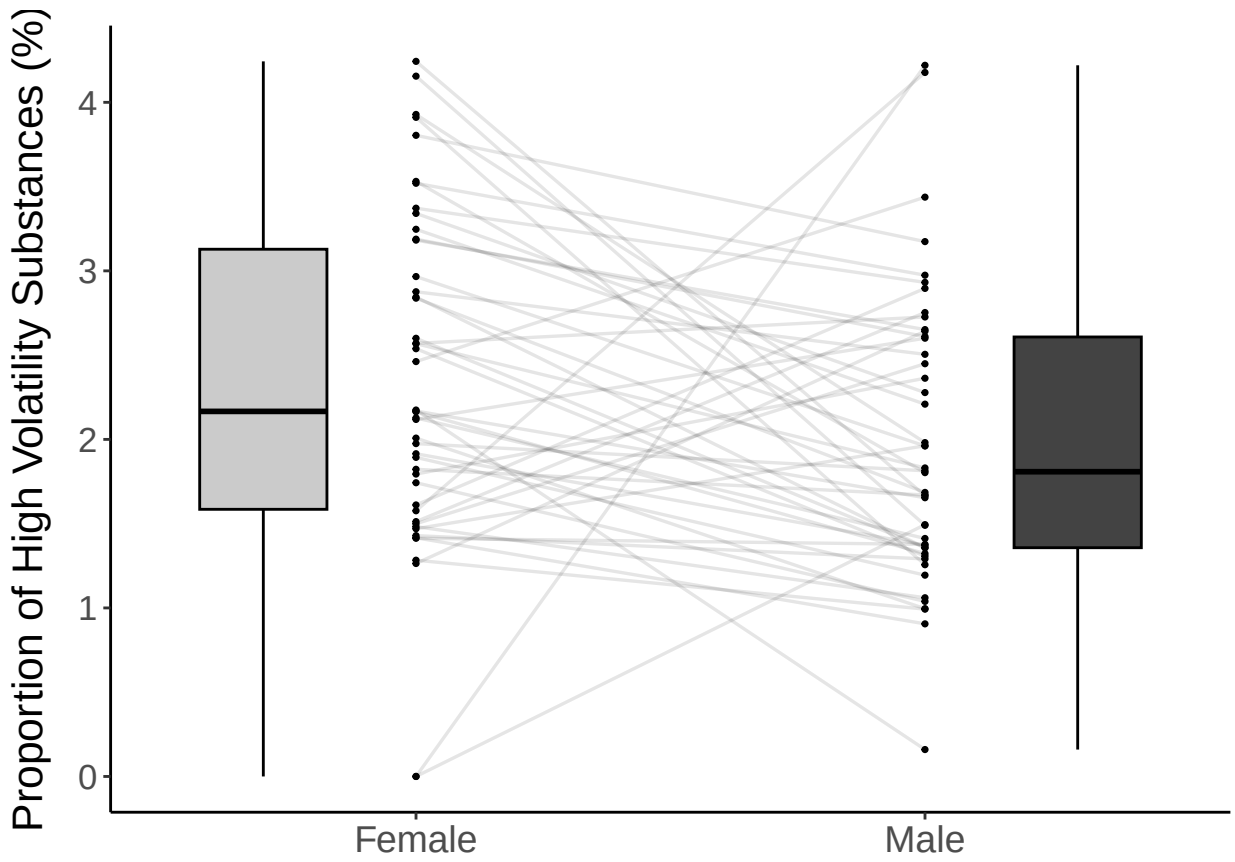
- Model diagnostic

```
check_model(LMM_Sex_D)
```



2.1.4 Volatility

a. Visualisation of the sex difference (within pairs) in terms of volatility.



b. Analysis

- Fitting the model

```
LMM_Sex_V <- lme4::lmer(formula = "Volatility ~ Sex + (1 | Nestbox_ID)",  data = Pairs_Nrearing)
summary(LMM_Sex_V)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Volatility ~ Sex + (1 | Nestbox_ID)
##    Data: Pairs_Nrearing
##
## REML criterion at convergence: 249.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5329 -0.7089 -0.1806  0.6859  2.4298
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.006745 0.08213
##  Residual               0.851410 0.92272
## Number of obs: 92, groups:  Nestbox_ID, 46
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)   2.3379     0.1366  17.117
## SexM         -0.3595     0.1924  -1.868
##
## Correlation of Fixed Effects:
##      (Intr)
## SexM -0.704
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Volatility ~ Sex + (1 | Nestbox_ID)",  data = Pairs_Nrearing))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Volatility ~ Sex + (1 | Nestbox_ID)"
##    Data: Pairs_Nrearing
##
## REML criterion at convergence: 249.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5329 -0.7089 -0.1806  0.6859  2.4298
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.006745 0.08213
##  Residual               0.851410 0.92272
## Number of obs: 92, groups:  Nestbox_ID, 46
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   2.3379     0.1366 89.9944  17.117   <2e-16 ***
## SexM         -0.3595     0.1924 45.0000  -1.868   0.0682 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## SexM -0.704
```

P-value = 0.068 –> non significant.

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_Sex_V, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term            estimate std.error stati~1 conf.~2 conf.~3
##   <chr>    <chr>      <chr>              <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>       (Intercept)       2.34       0.137   17.1    2.09   2.61
## 2 fixed    <NA>       SexM             -0.359      0.192   -1.87  -0.743 -0.0273
## 3 ran_pars Nestbox_ID sd__(Intercept)   0.0821    NA       NA      0      0.535
## 4 ran_pars Residual   sd__Observation   0.923     NA       NA      0.703  1.04
## # ... with abbreviated variable names 1: statistic, 2: conf.low, 3: conf.high
```

β estimate of sexM effect: -0.36 –> males have a proportion of on average 0.36 less volatile substances than females in our samples. Confidence interval: [-0.75 ; 0.048] –> includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_Sex_V, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.0367 2e-04    0.1385   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of sex: 0.0367 So the sex only explain 3.67% of the variation of volatility in our data.

- Finding the repeatability (random effect)

```
rpt(Volatility ~ Sex + (1 | Nestbox_ID),
    grname = "Nestbox_ID",
    data = Pairs_Nrearing, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

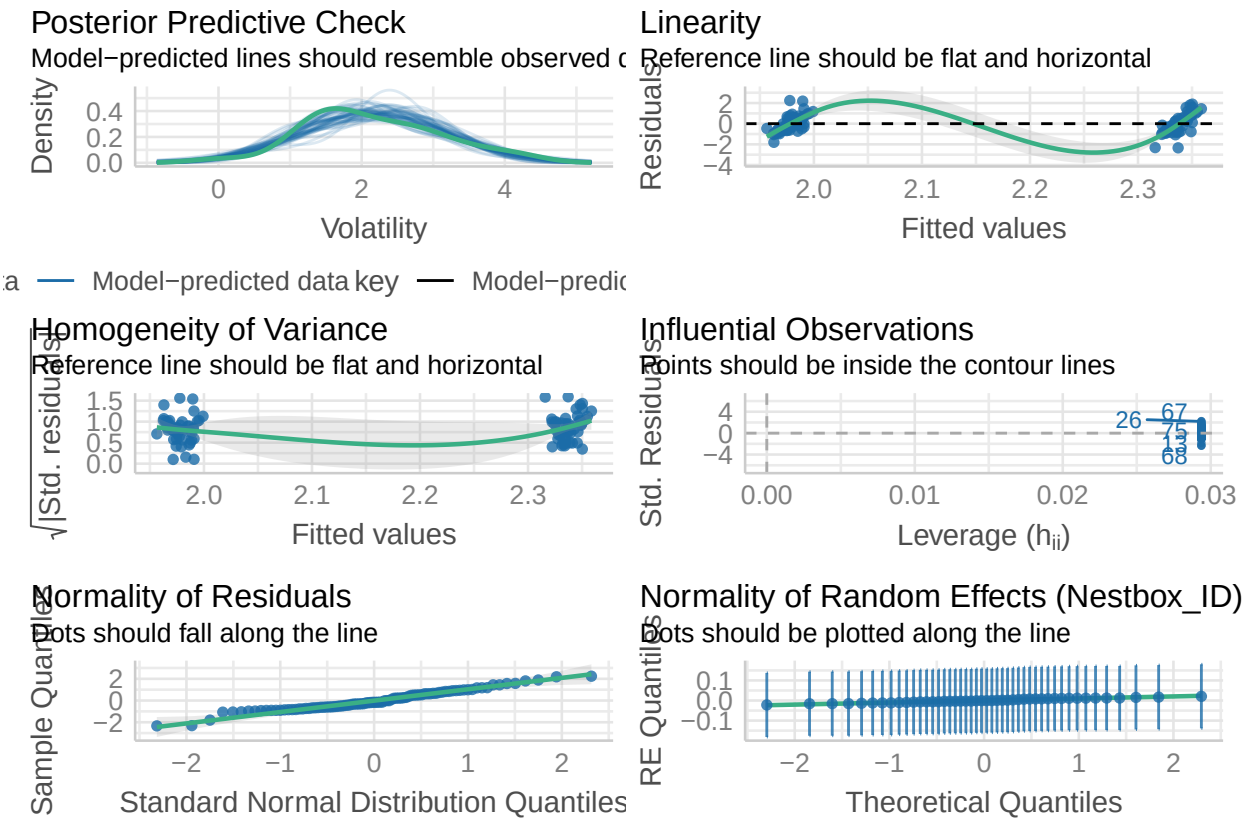
```
## Bootstrap Progress:
## Permutation Progress for Nestbox_ID :
```

```
##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Nestbox_ID
## R  = 0.008
## SE = 0.088
## CI = [0, 0.295]
## P  = 0.479 [LRT]
##      0.472 [Permutation]
```

Repeatability of the NestboxID effect: 0.008 –> 0.8% of the variation of volatility between samples is due to the variation between pairs in our data. P(perm)= 0.515, non significant. The volatility is not repeatable between partners.

- Model diagnostic

```
check_model(LMM_Sex_V)
```



2.2 Breeding stage and individual identity effects in females:

Data = 29 individual females (58 samples) sampled both during the incubation and nestling-rearing period

```
# Subset of the Metadata for the breeding stage and individual identity analyses:
F_sampled_twice <- Metadata%>%   # 58 samples
  filter(f_sampled_twice==1,
         Outliers==0,
         F_Connected_to_Outlier==0)%>%
  mutate(Breeding_Stage = as.factor(as.character(Breeding_Stage)))%>%
  select(-c(pair_brood, Partner_Connected_to_Outlier, Outliers, F_Connected_to_Outlier, Families, Famil
```

2.2.1 NMDS plot

Step 1: Building a Bray-Curtis matrix

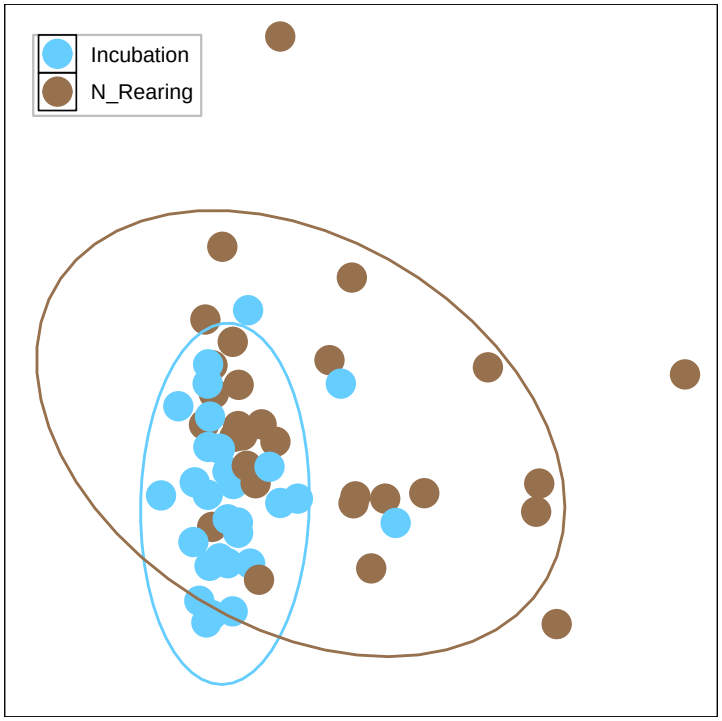
```
ChemdataBS <- Chemdata%>%
  filter(Sample %in% F_sampled_twice$Sample)%>%
  select(-Sample)
bc <- metaMDS(ChemdataBS, distance = "bray")
```

Step 2: Checking the stress

```
bc$stress
```

```
## [1] 0.07181659
```

Step 3: Plot the NMDS, here by breeding stage

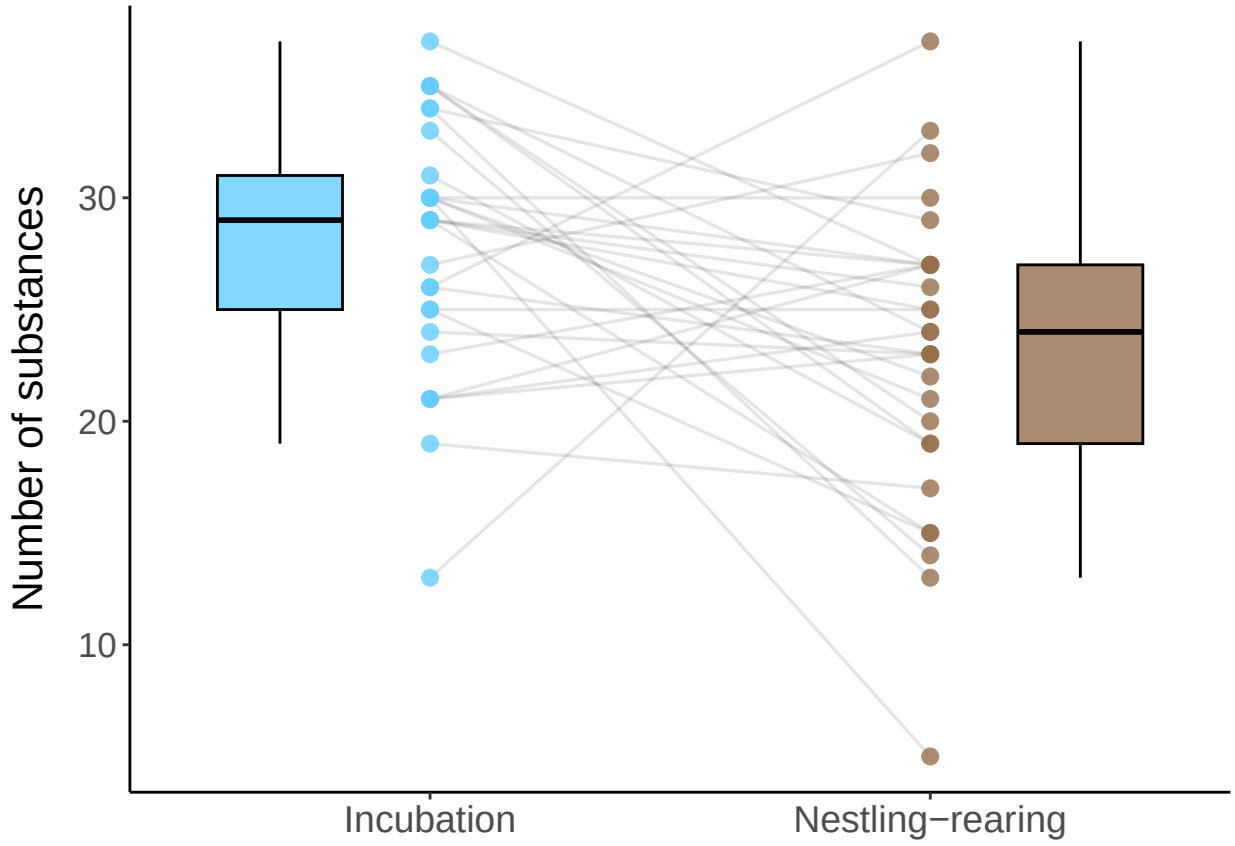


2.2.2 Richness

a. Visualisation of the breeding stage difference (within individuals) in terms of chemical richness.

```
theme_set(theme_classic())
ggplot(F_sampled_twice, aes(x=Breeding_Stage, y=Richness))+
  geom_boxplot(data = F_sampled_twice %>% filter(Breeding_Stage=="Incubation"),
               aes(x=Breeding_Stage, y=Richness),
               position=position_nudge(x=-0.3), outlier.shape = NA, width = 0.25,lwd = 0.5,
               alpha = 0.8, colour="black", fill = "#66cdff")+
  geom_boxplot(data = F_sampled_twice %>% filter(Breeding_Stage=="N_Rearing"), lwd = 0.5,
               aes(x=Breeding_Stage, y=Richness),
               position=position_nudge(x=0.3), outlier.shape = NA, width = 0.25,
               alpha = 0.8, colour="black", fill = "#97704d")+
  geom_line(aes(group=Individual_ID),alpha = 0.1, size = 0.6, position=position_dodge(0), show.legend =
  geom_point(aes(color = Breeding_Stage), alpha = 0.8, size=2.5,show.legend = FALSE)+
```

```
  scale_color_manual(values = c("#66cdff", "#97704d")) +
  scale_x_discrete(labels=c("Incubation","Nestling-rearing")) +
  ylab("Number of substances") +
  theme(axis.title.y=element_text(size=16, margin = margin(t = 0, r = 10, b = 0, l = 0)),
        axis.text.y = element_text(size=13),
        axis.text.x = element_text(size=14),
        axis.title.x = element_blank())
```



b. Analysis

- Fitting the model

```
LMM_B_Stage_R <- lme4::lmer(formula = "Richness ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampl
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(LMM_B_Stage_R)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Richness ~ Breeding_Stage + (1 | Individual_ID)
##    Data: F_sampled_twice
##
## REML criterion at convergence: 370.1
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9129 -0.6031  0.1668  0.6337  2.2458
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept)  0.00    0.000
##  Residual                  38.48    6.203
## Number of obs: 58, groups:  Individual_ID, 29
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)                27.966      1.152  24.278
## Breeding_StageN_Rearing    -4.897      1.629  -3.006
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Richness ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampled_tw
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Richness ~ Breeding_Stage + (1 | Individual_ID)"
##    Data: F_sampled_twice
##
## REML criterion at convergence: 370.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9129 -0.6031  0.1668  0.6337  2.2458
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept)  0.00    0.000
##  Residual                  38.48    6.203
## Number of obs: 58, groups:  Individual_ID, 29
##
## Fixed effects:
##                          Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)                27.966      1.152 56.000  24.278  < 2e-16 ***
## Breeding_StageN_Rearing    -4.897      1.629 56.000  -3.006  0.00396 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

P-value = 0.004 –> significant **

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_B_Stage_R, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group         term            estim~1 std.e~2 stati~3 conf.~4 conf.~5
##   <chr>    <chr>         <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>          (Intercept)       28.0     1.15   24.3    25.7    30.2
## 2 fixed    <NA>          Breeding_Stage~   -4.90    1.63   -3.01   -8.07   -1.85
## 3 ran_pars Individual_ID sd__(Intercept)    0      NA      NA       0       4.18
## 4 ran_pars Residual      sd__Observation    6.20   NA      NA       4.45    7.20
## # ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
## #   4: conf.low, 5: conf.high
```

β estimate of Nestling-rearing period effect: -4.90 –> females during nestling-rearing have on average 4.897 less substances than females during incubation in our samples. Confidence interval: [-8.25 ; -1.70] –> does not includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_B_Stage_R, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.1368 0.0172   0.3252   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of breeding stage: 0.1368 So the breeding stage explains 13.68% of the variation of richness in our data.

- Finding the repeatability (random effect)

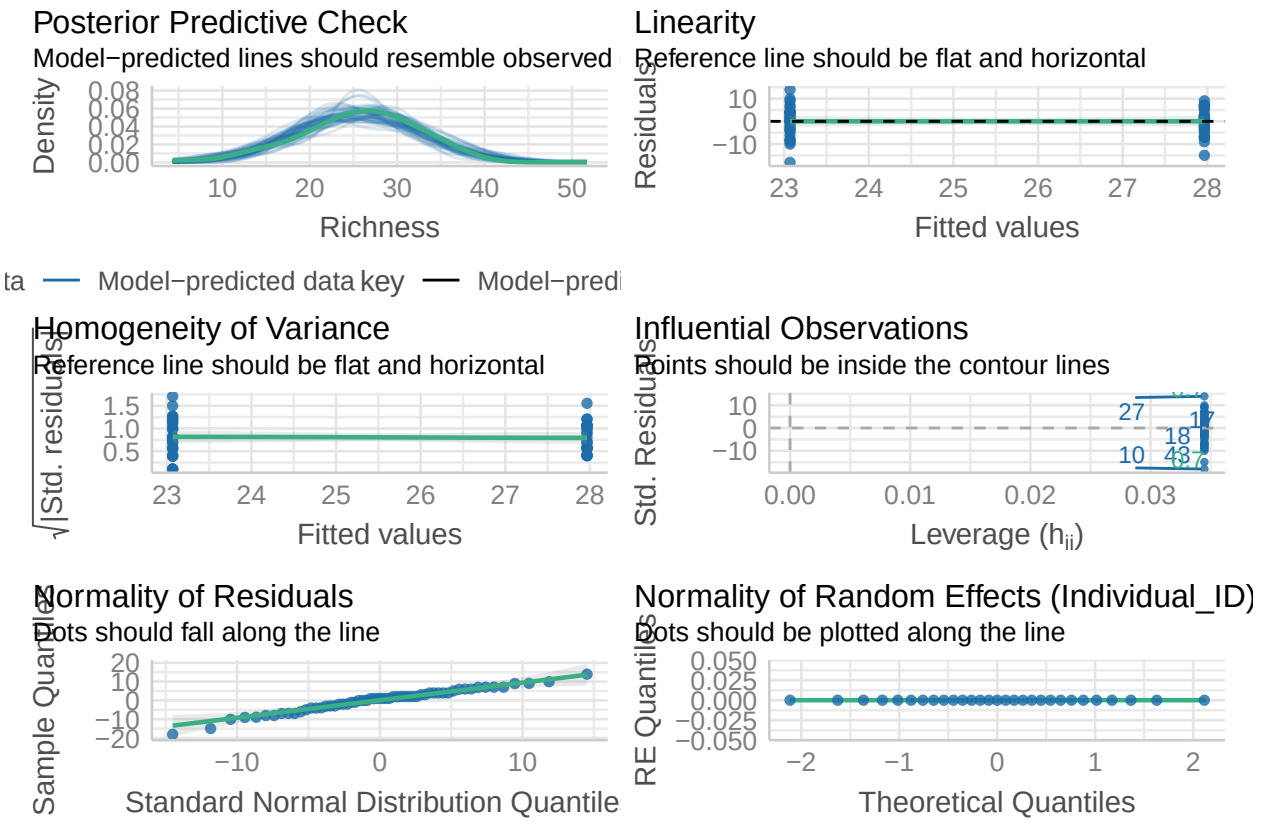
```
rpt(Richness ~ Breeding_Stage + (1 | Individual_ID),
    grname = "Individual_ID",
    data = F_sampled_twice, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Individual_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Individual_ID
## R  = 0
## SE = 0.118
## CI = [0, 0.384]
## P  = 0.5 [LRT]
##      1 [Permutation]
```

Repeatability of the Individual ID effect: 0 –> 0% of the variation of richness between samples is due to the variation between individuals in our data. The chemical richness is not repeatable among individuals.

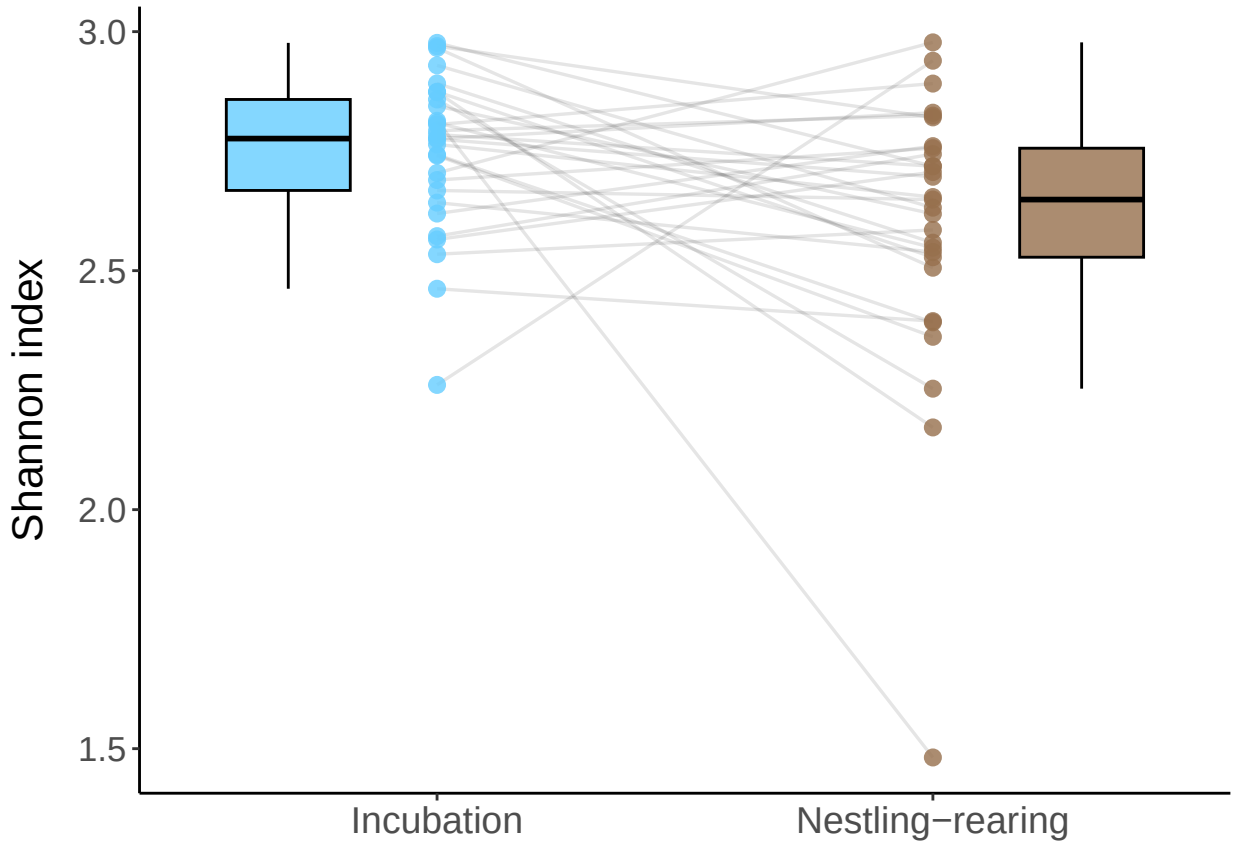
- Model diagnostic

```
check_model(LMM_B_Stage_R)
```



2.2.3 Diversity

a. Visualisation of the breeding stage difference (within individuals) in terms of Shannon diversity.



b. Analysis

- Fitting the model

```
LMM_B_Stage_D <- lme4::lmer(formula = "Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)",  data = F
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(LMM_B_Stage_D)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)
##    Data: F_sampled_twice
##
## REML criterion at convergence: 3.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7481 -0.3296  0.1555  0.5355  1.6312
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.000    0.0000
##  Residual                  0.055    0.2345
```

```
## Number of obs: 58, groups:  Individual_ID, 29
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)               2.74851    0.04355  63.115
## Breeding_StageN_Rearing -0.15344    0.06159  -2.492
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampl
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)"
##    Data: F_sampled_twice
##
## REML criterion at convergence: 3.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7481 -0.3296  0.1555  0.5355  1.6312
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.000    0.0000
##  Residual                  0.055    0.2345
## Number of obs: 58, groups:  Individual_ID, 29
##
## Fixed effects:
##                          Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)               2.74851    0.04355 56.00000  63.115   <2e-16 ***
## Breeding_StageN_Rearing -0.15344    0.06159 56.00000  -2.492   0.0157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

P-value = 0.0157 –> significant (*)

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_B_Stage_D, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group         term            estim~1 std.e~2 stati~3 conf.~4 conf.~5
##   <chr>    <chr>         <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>          (Intercept)       2.75   0.0435   63.1    2.65   2.83
## 2 fixed    <NA>          Breeding_Stage~  -0.153  0.0616   -2.49  -0.271 -0.0236
## 3 ran_pars Individual_ID sd__(Intercept)   0     NA        NA      0      0.149
## 4 ran_pars Residual      sd__Observation   0.235 NA        NA      0.172  0.278
## # ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
## #   4: conf.low, 5: conf.high
```

β estimate of Nestling-rearing period effect: -0.153 –> females during nestling-rearing have on average 0.153 less diversity than females during incubation in our samples. Confidence interval: [-0.268; -0.0479] –> does not includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_B_Stage_D, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.0982 0.0059   0.2745   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of breeding stage: 0.0982 So the breeding stage explains 9.82% of the variation of the diversity in our data.

- Finding the repeatability (random effect)

```
rpt(Shannon_Index ~ Breeding_Stage + (1 | Individual_ID),
    grname = "Individual_ID",
    data = F_sampled_twice, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

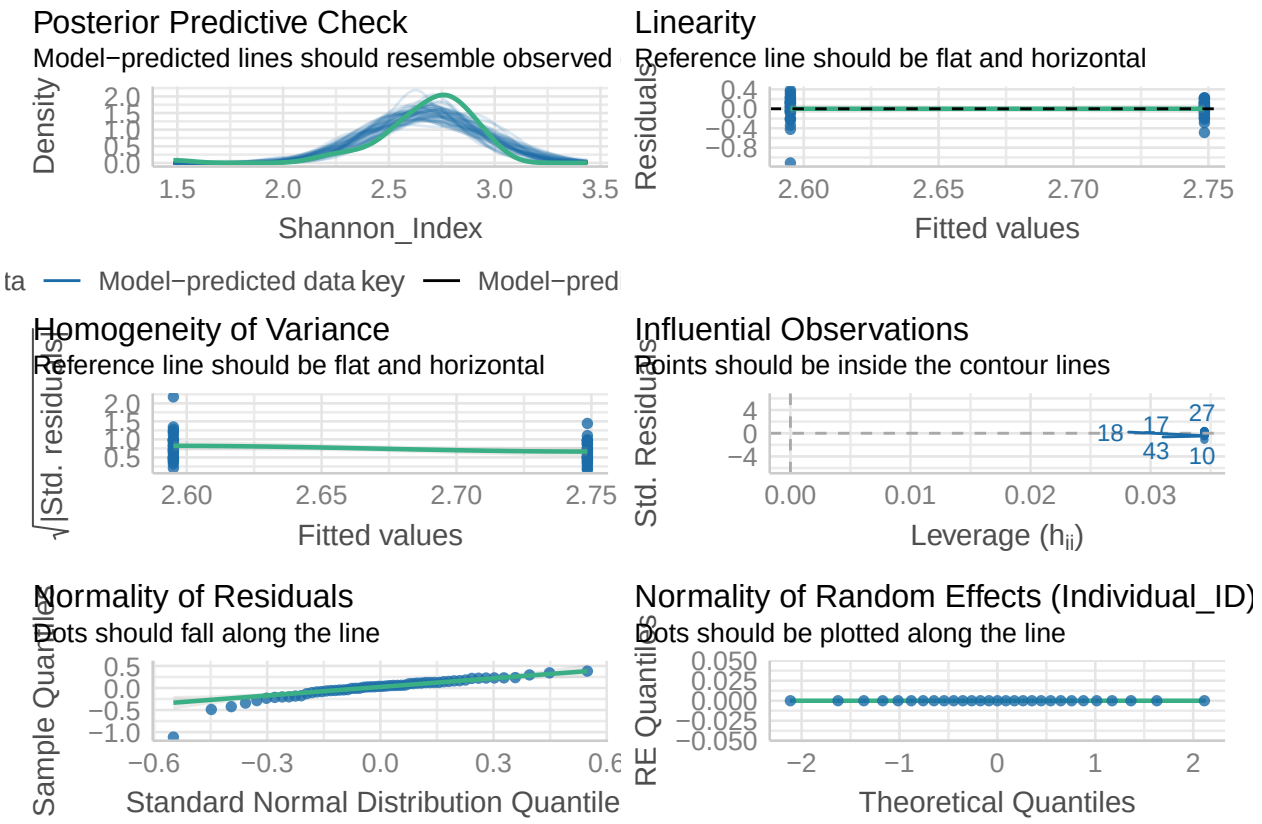
```
## Bootstrap Progress:
## Permutation Progress for Individual_ID :

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

```
## R  = 0
## SE = 0.113
## CI = [0, 0.37]
## P  = 1 [LRT]
##      1 [Permutation]
```

Repeatability of the Individual ID effect: 0 –> 0% of the variation of diversity between samples is due to the variation between individuals in our data. The diversity is not repeatable among individuals.

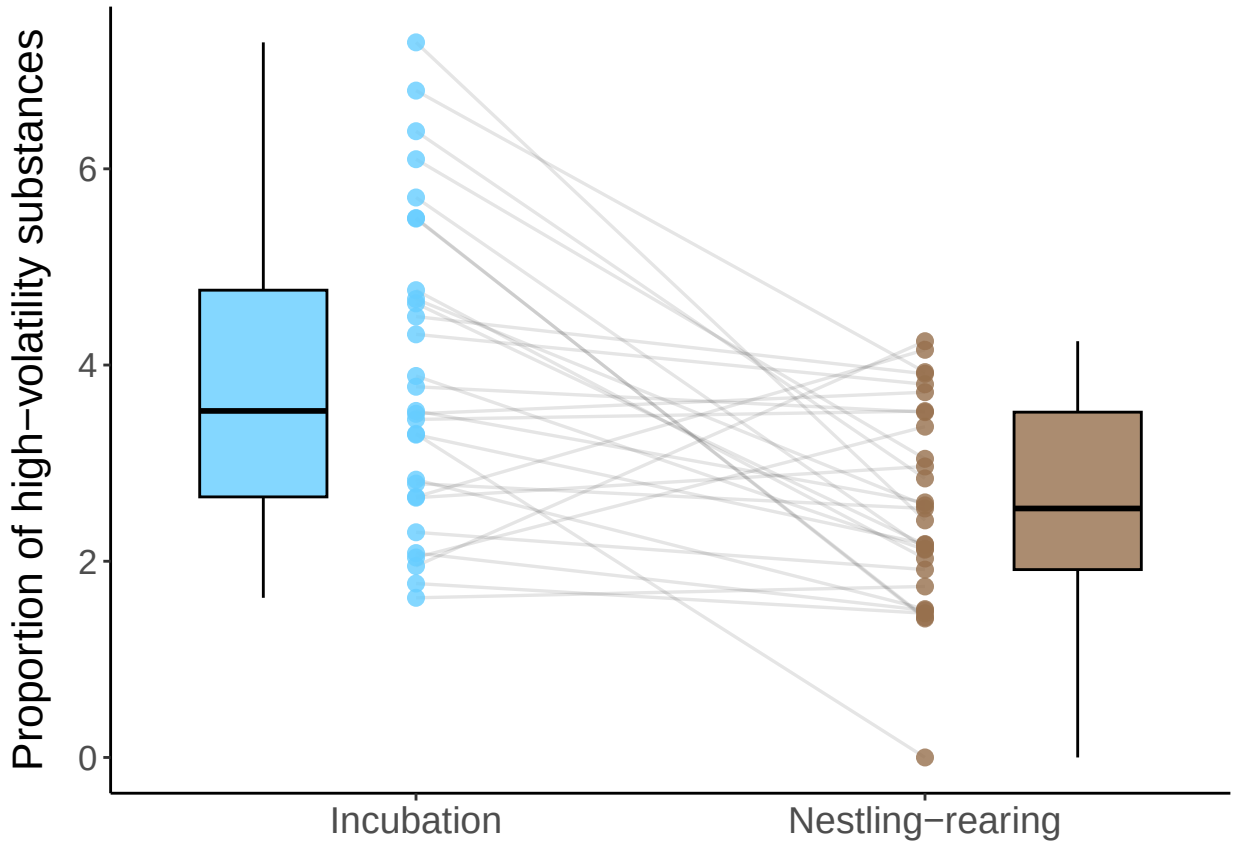
- Model diagnostic

```
check_model(LMM_B_Stage_D)
```



2.2.4 Volatility

a. Visualisation of the breeding stage difference (within individuals) in terms of volatility.



b. Analysis

- Fitting the model

```
LMM_B_Stage_V <- lme4::lmer(formula = "Volatility ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sa
summary(LMM_B_Stage_V)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Volatility ~ Breeding_Stage + (1 | Individual_ID)
##    Data: F_sampled_twice
##
## REML criterion at convergence: 198.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8401 -0.7040 -0.2028  0.6813  2.4513
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.1125   0.3355
##  Residual                  1.6872   1.2989
## Number of obs: 58, groups:  Individual_ID, 29
##
## Fixed effects:
##                         Estimate Std. Error t value
```

```
## (Intercept)               3.9162     0.2491  15.720
## Breeding_StageN_Rearing  -1.3380     0.3411  -3.922
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.685
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Volatility ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampled_
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Volatility ~ Breeding_Stage + (1 | Individual_ID)"
##    Data: F_sampled_twice
##
## REML criterion at convergence: 198.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8401 -0.7040 -0.2028  0.6813  2.4513
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.1125   0.3355
##  Residual                  1.6872   1.2989
## Number of obs: 58, groups:  Individual_ID, 29
##
## Fixed effects:
##                          Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)               3.9162     0.2491 55.7819  15.720  < 2e-16 ***
## Breeding_StageN_Rearing  -1.3380     0.3411 28.0000  -3.922 0.000517 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.685
```

P-value = 0.0005 –> significant ***

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_B_Stage_V, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group         term            estim~1 std.e~2 stati~3 conf.~4 conf.~5
##   <chr>    <chr>         <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>          (Intercept)       3.92    0.249   15.7    3.44    4.38
## 2 fixed    <NA>          Breeding_Stage~  -1.34    0.341   -3.92  -2.05   -0.714
## 3 ran_pars Individual_ID sd__(Intercept)   0.335  NA       NA      0       0.929
## 4 ran_pars Residual      sd__Observation   1.30   NA       NA      0.953   1.57
## # ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
## #   4: conf.low, 5: conf.high
```

β estimate of Nestling-rearing period effect: -0.013 –> females during nestling-rearing have a proportion of on average 0.013 less diversity than females during incubation in our samples. Confidence interval: [-0.020 ; -0.006] –> does not includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_B_Stage_V, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.2019 0.0577   0.3795   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of breeding stage: 0.2019 So the breeding stage explains 20.2% of the variation of the volatility in our data.

- Finding the repeatability (random effect)

```
rpt(Volatility ~ Breeding_Stage + (1 | Individual_ID),
    grname = "Individual_ID",
    data = F_sampled_twice, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

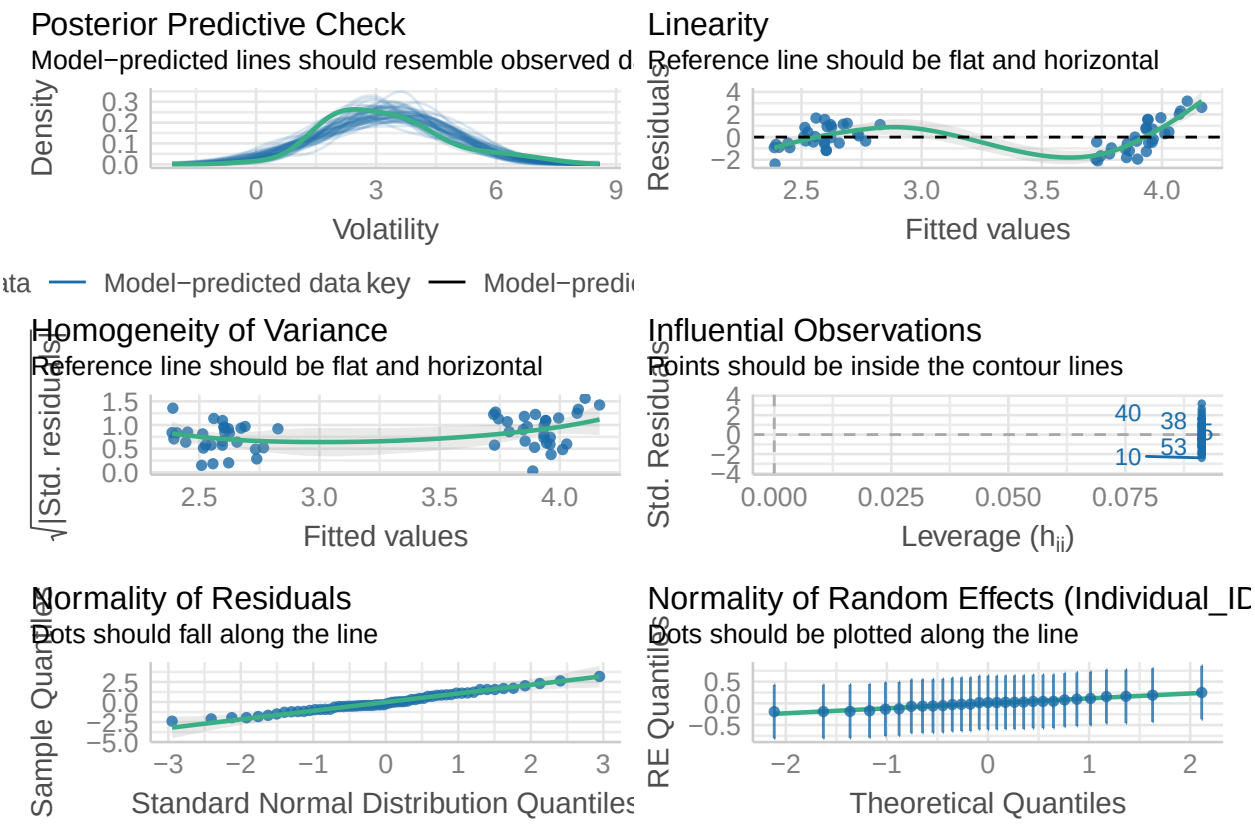
```
## Bootstrap Progress:
## Permutation Progress for Individual_ID :
```

```
##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Individual_ID
## R  = 0.063
## SE = 0.132
## CI = [0, 0.433]
## P  = 0.368 [LRT]
##      0.392 [Permutation]
```

Repeatability of the Individual ID effect: 0.063 –> 6.3% of the variation of volatility between samples is due to the variation between individuals in our data. P(perm)= 0.411, non significant. The volatility is not repeatable among individuals.

- Model diagnostic

```
check_model(LMM_B_Stage_V)
```



3 Effect sizes (comparisons with pilot paper)

To compare the results from the replication study and the original study, we calculated effect sizes for the LMM analyses (chemical richness, Shannon diversity and volatility). For fixed effects, we calculated a corrected version of the standardised effect size Cohen's d (Cohen 1987) for small sample sizes, the Hedges' g (Hedges & Olkin, 1985) and its 95% confidence interval, using the *effsize* package (Torchiano & Torchiano, 2020). We used the repeatabilities with their confidence interval as effect sizes for random effects (Stoffel et al. 2017).

3.1 Calculation of Hedges'g effect sizes for both studies

```
#### Sex effect
#Original study
cohen.d(Richness ~ sex, data=Pairs_NrearingP1, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.1424218 (negligible)
```

```
## 95 percent confidence interval:
##      lower      upper
## -0.7143432  0.9991869
```

```
#Replication study
cohen.d(Richness ~ Sex, data=Pairs_Nrearing, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: -0.1833413 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.5949919  0.2283094
```

```
#Original study
cohen.d(Diversity ~ sex, data=Pairs_NrearingP1, hedges.correction=TRUE) #Replication study
```

```
##
## Hedges's g
##
## g estimate: 0.5185992 (medium)
## 95 percent confidence interval:
##     lower     upper
## -0.351346  1.388544
```

```
cohen.d(Shannon_Index ~ Sex, data=Pairs_Nrearing, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: -0.1576961 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.5691226  0.2537305
```

```
#Original study
cohen.d(Volatility ~ sex, data=Pairs_NrearingP1, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.9643663 (large)
## 95 percent confidence interval:
##      lower      upper
## 0.06031588 1.86841668
```

```
#Replication study
cohen.d(Volatility ~ Sex, data=Pairs_Nrearing, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.3848055 (small)
## 95 percent confidence interval:
##       lower       upper
## -0.02976734  0.79937840
```

```
### Breeding stage effect
#Original study
cohen.d(Richness ~ breeding_stage, data=F_sampled_twiceP1, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.05244766 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.9616258  1.0665211
```

```
#Replication study
cohen.d(Richness ~ Breeding_Stage, data=F_sampled_twice, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.7787462 (medium)
## 95 percent confidence interval:
##     lower     upper
## 0.2404344 1.3170579
```

```
#Original study
cohen.d(Diversity ~ breeding_stage, data=F_sampled_twiceP1, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.285192 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.7338482  1.3042322
```

```
#Replication study
cohen.d(Shannon_Index ~ Breeding_Stage, data=F_sampled_twice, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.6455071 (medium)
## 95 percent confidence interval:
##     lower     upper
## 0.1131631 1.1778511
```

```r
#Original study
cohen.d(Volatility ~ breeding_stage, data=F_sampled_twiceP1, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: -0.4771426 (small)
## 95 percent confidence interval:
##      lower      upper
## -1.5053674  0.5510822
```

```r
#Replication study
cohen.d(Volatility ~ Breeding_Stage, data=F_sampled_twice, hedges.correction=TRUE)
```

```
##
## Hedges's g
##
## g estimate: 0.9839316 (large)
## 95 percent confidence interval:
##     lower     upper
## 0.4344252 1.5334379
```

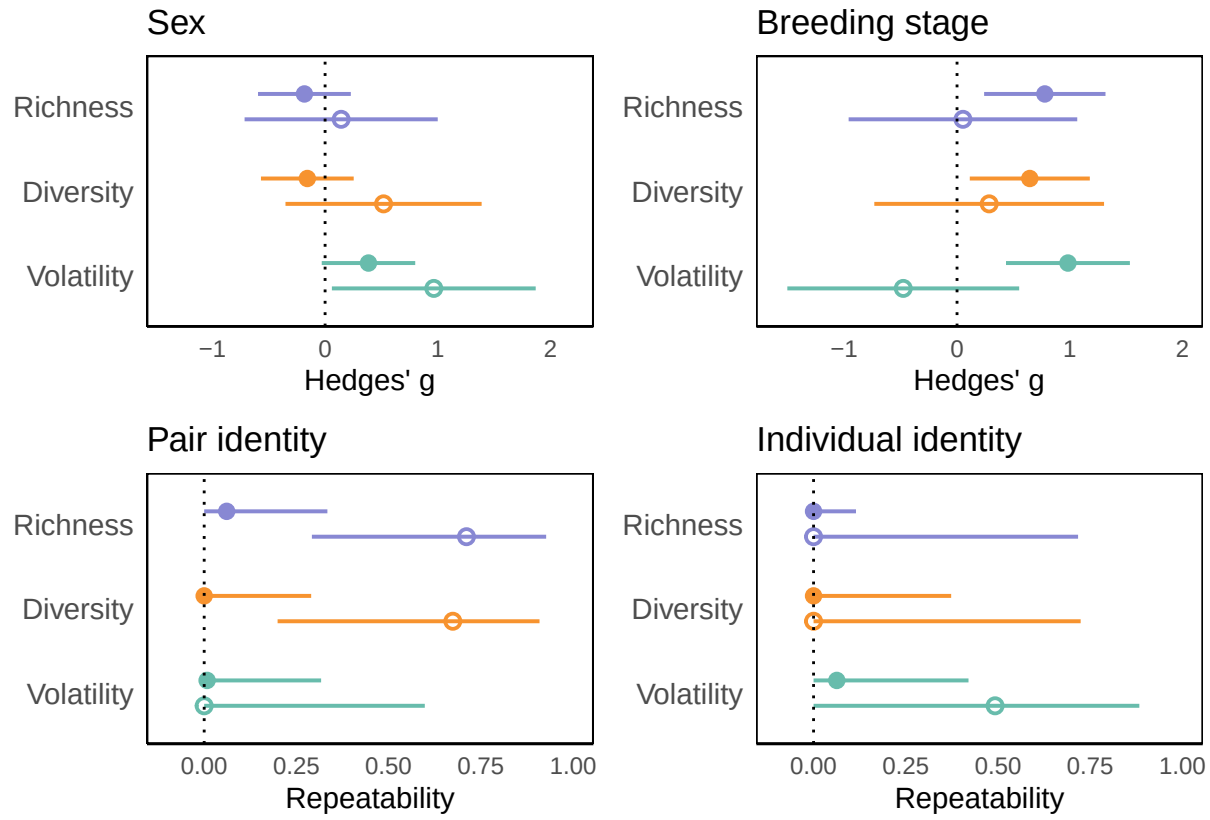
3.2 Plot

The values obtained with the cohen.d() function were manually collected into an exel file, which we will use now to plot the results

```r
eff_sizes <- eff_sizes %>%
  mutate(Paper=as.character(as.double(Paper)),
    Response = case_when(
    Response == "Shannon_Diversity" ~ "Diversity",
    TRUE ~ Response
  ))
# subset of the data for Sex and Richness, Diversity, and Volatility in paper 1 and 2
subset_sex <- eff_sizes[eff_sizes$Effect == "Sex" & eff_sizes$Response %in% c("Richness", "Diversity",

# subset of the data for BreedingStage and Richness, Diversity, and Volatility in paper 1 and 2
subset_breeding <- eff_sizes[eff_sizes$Effect == "BreedingStage" & eff_sizes$Response %in% c("Richness"

# create a new variable for interaction grouping
subset_sex$Interaction <- interaction(subset_sex$Effect, subset_sex$Response, subset_sex$Paper, sep = "
subset_breeding$Interaction <- interaction(subset_breeding$Effect, subset_breeding$Response, subset_bre

# reorder levels
subset_sex$Response <- factor(subset_sex$Response, levels = c("Richness", "Diversity", "Volatility"))
subset_breeding$Response <- factor(subset_breeding$Response, levels = c("Richness", "Diversity", "Volat

# colours
col_response <- c("Richness" = "#8888D3",
                  "Diversity" = "#F7932F",
                  "Volatility" = "#68BCAC")
```

```r
# Plot Sex
sex <- ggplot(subset_sex, aes(x = fct_rev(Response), y = Estimate, color = Response, shape = Paper, size
  geom_pointrange(aes(ymin = int.inf, ymax = int.sup), position = position_dodge(width = 0.6), linewidth
  geom_hline(yintercept = 0, linetype = "dotted", color = "black") +
  labs(title = "Sex", y = "Hedges' g") +
  ylim(-1.4, 2.2) +
  theme_minimal() +
  coord_flip() +
  scale_color_manual(values = col_response) +
  theme(axis.title.y = element_blank(), legend.position = "none", panel.grid = element_blank(),
        axis.line.x = element_line(), panel.border = element_rect(fill = NA),
        axis.text.y = element_text(size = 11)) +
  scale_shape_manual(values = c(1, 16))+
  scale_size_manual(values = c(0.5,0.6))

# Plot Breeding stage
breeding <- ggplot(subset_breeding, aes(x = fct_rev(Response), y = Estimate, color = Response, shape = P
  geom_pointrange(aes(ymin = int.inf, ymax = int.sup), position = position_dodge(width = 0.6), linewidth
  geom_hline(yintercept = 0, linetype = "dotted", color = "black") +
  labs(title = "Breeding stage", y = "Hedges' g") +
  ylim(-1.6, 2.0) +
  theme_minimal() +
  coord_flip() +
  scale_color_manual(values = col_response) +
  theme(axis.title.y = element_blank(), legend.position = "none", panel.grid = element_blank(),
        axis.line.x = element_line(), panel.border = element_rect(fill = NA),
        axis.text.y = element_text(size = 11)) +
  scale_shape_manual(values = c(1, 16))+
  scale_size_manual(values = c(0.5,0.6))

# Plot Pair ID
pair <- ggplot(subset_sex, aes(x = fct_rev(Response), y = Repeatability, color = Response, shape = Paper
  geom_pointrange(aes(ymin = Reap.Int.inf, ymax = Reap.Int.sup), position = position_dodge(width = 0.6)
  geom_hline(yintercept = 0, linetype = "dotted", color = "black") +
  labs(title = "Pair identity", y = "Repeatability") +
  ylim(-0.1, 1) +
  theme_minimal() +
  coord_flip() +
  scale_color_manual(values = col_response) +
  theme(axis.title.y = element_blank(), legend.position = "none", panel.grid = element_blank(),
        axis.line.x = element_line(), panel.border = element_rect(fill = NA),
        axis.text.y = element_text(size = 11)) +
  scale_shape_manual(values = c(1, 16))+
  scale_size_manual(values = c(0.5,0.6))

# Plot Individual ID
individual <- ggplot(subset_breeding, aes(x = fct_rev(Response), y = Repeatability, color = Response, sh
  geom_pointrange(aes(ymin = Reap.Int.inf, ymax = Reap.Int.sup), position = position_dodge(width = 0.6)
  geom_hline(yintercept = 0, linetype = "dotted", color = "black") +
  labs(title = "Individual identity", y = "Repeatability") +
  ylim(-0.1, 1) +
  theme_minimal() +
  coord_flip() +
```

```
  scale_color_manual(values = col_response) +
  theme(axis.title.y = element_blank(), legend.position = "none", panel.grid = element_blank(),
        axis.line.x = element_line(), panel.border = element_rect(fill = NA),
        axis.text.y = element_text(size = 11)) +
  scale_shape_manual(values = c(1, 16))+
  scale_size_manual(values = c(0.5,0.6))

# Combine the plots using patchwork
final <- sex + breeding + pair + individual + plot_layout(ncol = 2)
final
```



In this figure, we can see that the confidence intervals around the effect sizes of the GLMM results from the pilot study always overlap the confidence intervals for the effect sizes of the replication. This could mean that the same effect is being tested in both studies, which is what is desired. However, it should be noted that the confidence intervals around the effect sizes of the pilot study are especially wide, therefore making it easier for each pair of intervals to overlap over one another.

4 Exploratory analysis

4.1 Spatial analysis of pairs during nestling-rearing:

Gilles et al. (2024) found a high similarity in preen oil composition between breeding partners and proposed that this may be due to their spatial proximity, as they share the same territory and the same food available. To test for the effect of spatial proximity on preen oil composition, we ran Mantel tests of the spatial versus the Bray-Curtis distance, along with Mantel correlograms (Borcard et al. 2011) and scatterplots for

visualisation, using the *vegan* package (Oksanen et al. 2010) in R. This method tests whether chemical similarity covaries with spatial proximity by comparing pairwise chemical distances with pairwise spatial distances. We used all the samples from adult males and females during nestling-rearing for which we had the GPS position of the nestbox (regardless of whether they were part of a complete breeding pair). We tested males (N=42) and females (N=44) separately to control for the effect of breeding partner proximity.

4.1.1 Spatial analysis in females:

Preparation of the data

```
# Select all samples from females during nestling-rearing (50 samples)
Females_Nrearing <-Metadata%>%
  filter(Ageclass=="A",
         Breeding_Stage == "N_Rearing",
         Sex == "F",
         Outliers == 0,
         Sample != "L280")%>%
  select(-c(Richness, Shannon_Index, Volatility, pair_brood, f_sampled_twice, Families, FamiliesOrdered

# Select the GPS data for the nestboxes in which we sampled females during nestling-rearing
FGPSNetboxes <- GPSNetboxes%>%
  filter(Name %in% Females_Nrearing$Nestbox_ID)%>%
  rename(Nestbox_ID=Name)
# Select samples from females for which we have GPS data on the nestbox (N = 44 samples)
Females_Nrearing <- Females_Nrearing%>%
  filter(Nestbox_ID %in% FGPSNetboxes$Nestbox_ID)
df_list <- list(Females_Nrearing, FGPSNetboxes)
Females_Nrearing <- df_list%>%
  reduce(full_join, by="Nestbox_ID") # Combine it in a single data frame

# Select the chemical data from the subset of samples
Chem_F <- Chemdata%>%
  filter(Sample %in% Females_Nrearing$Sample)
Chem_F <- subset(Chem_F, select = !apply(Chem_F, 2, function(x) all(x == 0))) # to remove columns where
Females_Nrearing <- list(Females_Nrearing, Chem_F)%>%
  reduce(full_join, by="Sample") # Add the chemical data to the main data frame
Chem_F <- Females_Nrearing%>%
  select(-c(1:15)) # Chemical data file in the right order
CoordinatesF <- Females_Nrearing%>% #Coordinates data file in the right order
  select(c(Easting, Northing))%>%
  rename(x=Easting, y=Northing)
Chem_F_det <- resid(lm(as.matrix(Chem_F)~., data=CoordinatesF)) # detrend Chem_F data
ChemF_det_1 <- Chem_F_det + 1 # we add 1 the the detrend data, because in order to create the bray-curt
MatrixF <- vegdist(ChemF_det_1) # Our Bray-curtis distance (dissimilarity) matrix
```

Mantel test:

```
# For the Mantel test, we need to build the matrix of spatial distances:
SpatialMatrixF <- dist(CoordinatesF)

mantel(MatrixF, SpatialMatrixF, permutations=1000)
```

```
##
```

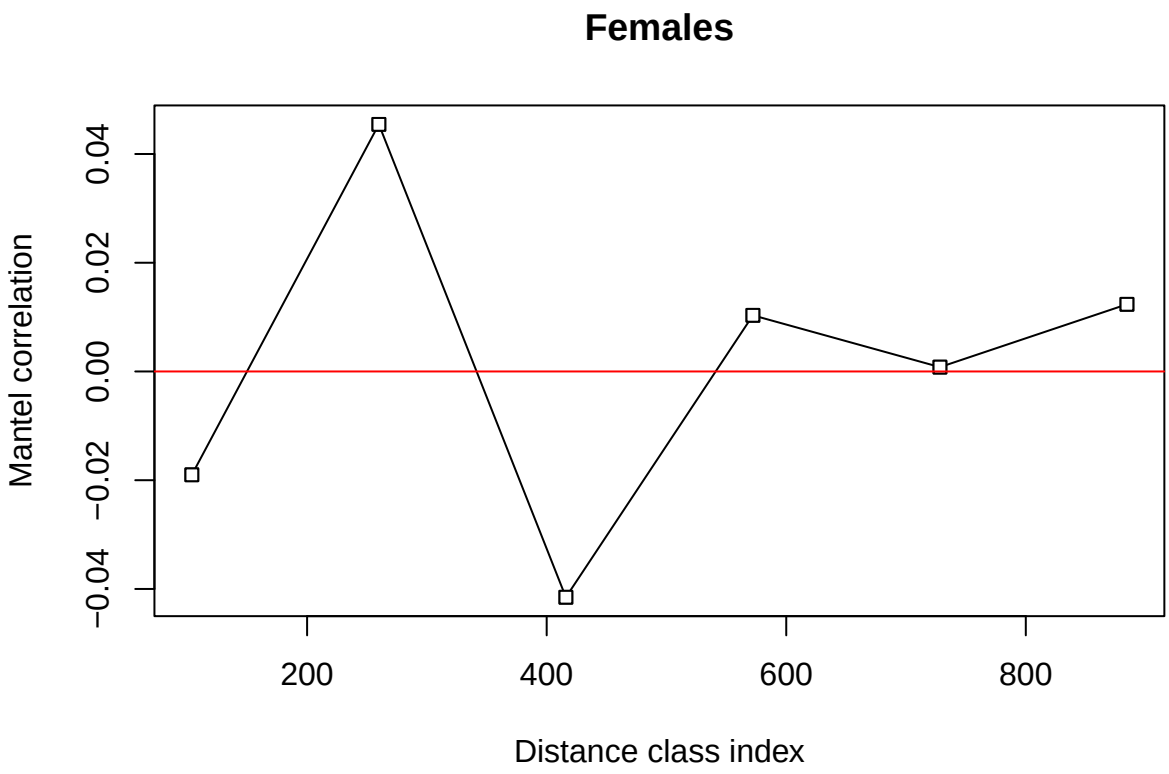
```
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = MatrixF, ydis = SpatialMatrixF, permutations = 1000)
##
## Mantel statistic r: -0.007228
##       Significance: 0.51948
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0947 0.1223 0.1566 0.1776
## Permutation: free
## Number of permutations: 1000
```

Mantel correlogram:

```
Mantel_correlog_F <- mantel.correlog(MatrixF, XY=CoordinatesF, nperm=9999)
summary(Mantel_correlog_F)
```

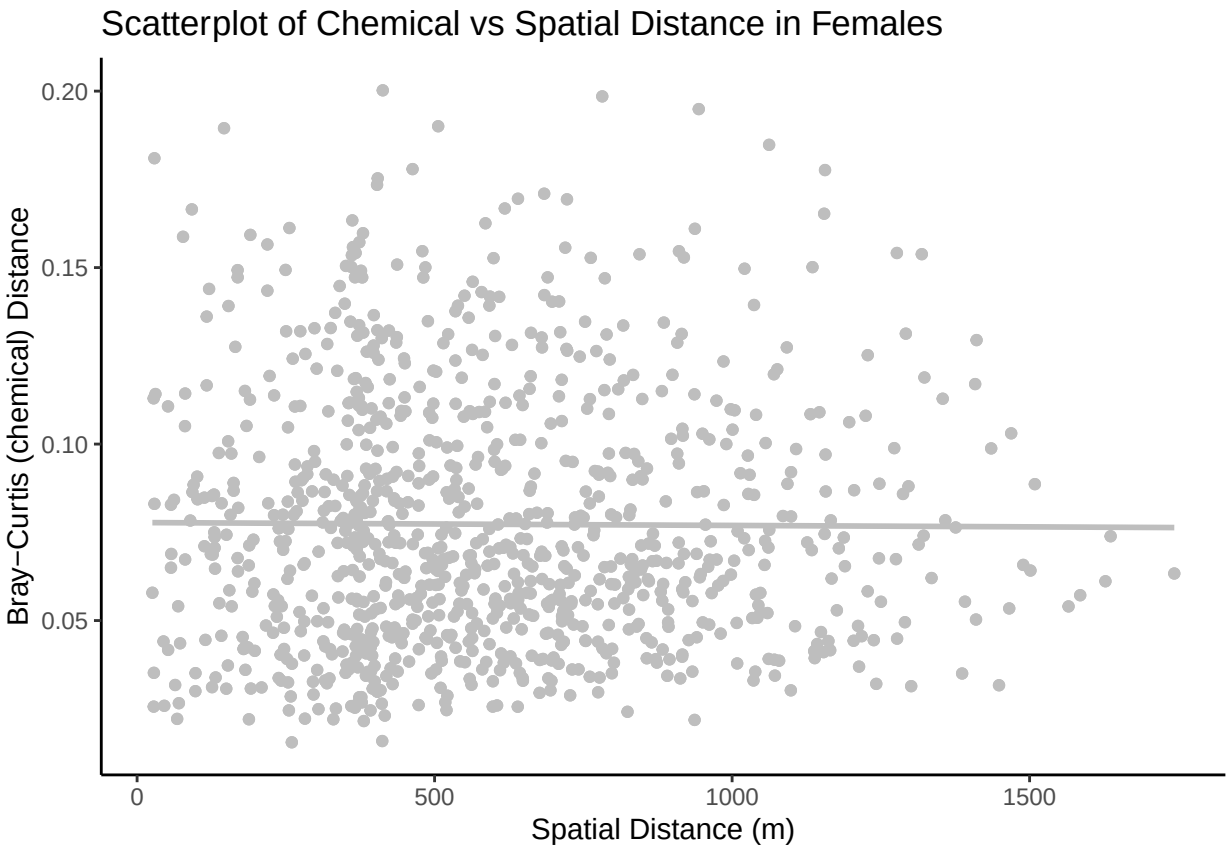
```
##            Length Class  Mode
## mantel.res 55     -none- numeric
## n.class     1     -none- numeric
## break.pts  12     -none- numeric
## mult        1     -none- character
## n.tests     1     -none- numeric
## call        4     -none- call
```

```
FBC <- plot(Mantel_correlog_F)
title(main="Females")
```



Scatterplot:

```
ScatterF <- data.frame(SpatialMatrixF = as.vector(as.matrix(SpatialMatrixF)),
                       MatrixF = as.vector(as.matrix(MatrixF)))%>%
  filter(MatrixF != 0)
ggplot(ScatterF, aes(x=SpatialMatrixF, y=MatrixF))+
  geom_point(color="grey") +
  geom_smooth(method = "lm", se = FALSE, color = "grey") +
  labs(x = "Spatial Distance (m)", y = "Bray-Curtis (chemical) Distance")+
  ggtitle("Scatterplot of Chemical vs Spatial Distance in Females")
```



4.1.2 Spatial analysis in males:

Preparation of the data

```
Males_Nrearing <-Metadata%>%
  filter(Ageclass=="A",
         Breeding_Stage == "N_Rearing",
         Sex == "M",
         Outliers == 0)%>%
  select(-c(Richness, Shannon_Index, Volatility, pair_brood, f_sampled_twice, Families, FamiliesOrdered

MGPSNetboxes <- GPSNetboxes%>%
  filter(Name %in% Males_Nrearing$Nestbox_ID)%>%
  rename(Nestbox_ID=Name)
Males_Nrearing <- Males_Nrearing%>%
  filter(Nestbox_ID %in% MGPSNetboxes$Nestbox_ID)
df_list <- list(Males_Nrearing, MGPSNetboxes)
Males_Nrearing <- df_list%>%
  reduce(full_join, by="Nestbox_ID")

Chem_M <- Chemdata%>%
  filter(Sample %in% Males_Nrearing$Sample)
Chem_M <- subset(Chem_M, select = !apply(Chem_M, 2, function(x) all(x == 0)))
Males_Nrearing <- list(Males_Nrearing, Chem_M)%>%
  reduce(full_join, by="Sample")
```

```
Chem_M <- Males_Nrearing%>%
  select(-c(1:15))
CoordinatesM <- Males_Nrearing%>%
  select(c(Easting, Northing))%>%
  rename(x=Easting, y=Northing)
Chem_M_det <- resid(lm(as.matrix(Chem_M)~., data=CoordinatesM)) #detrend matrixF
Chem_M_det_1 <- Chem_M_det + 1
MatrixM <- vegdist(Chem_M_det_1) #BC Matrix of detrend data
```

Mantel test:

```
# For the Mantel test, we need to build the matrix of spatial distances:
SpatialMatrixM <- dist(CoordinatesM, diag=T)

mantel(MatrixM, SpatialMatrixM, permutations=1000)
```

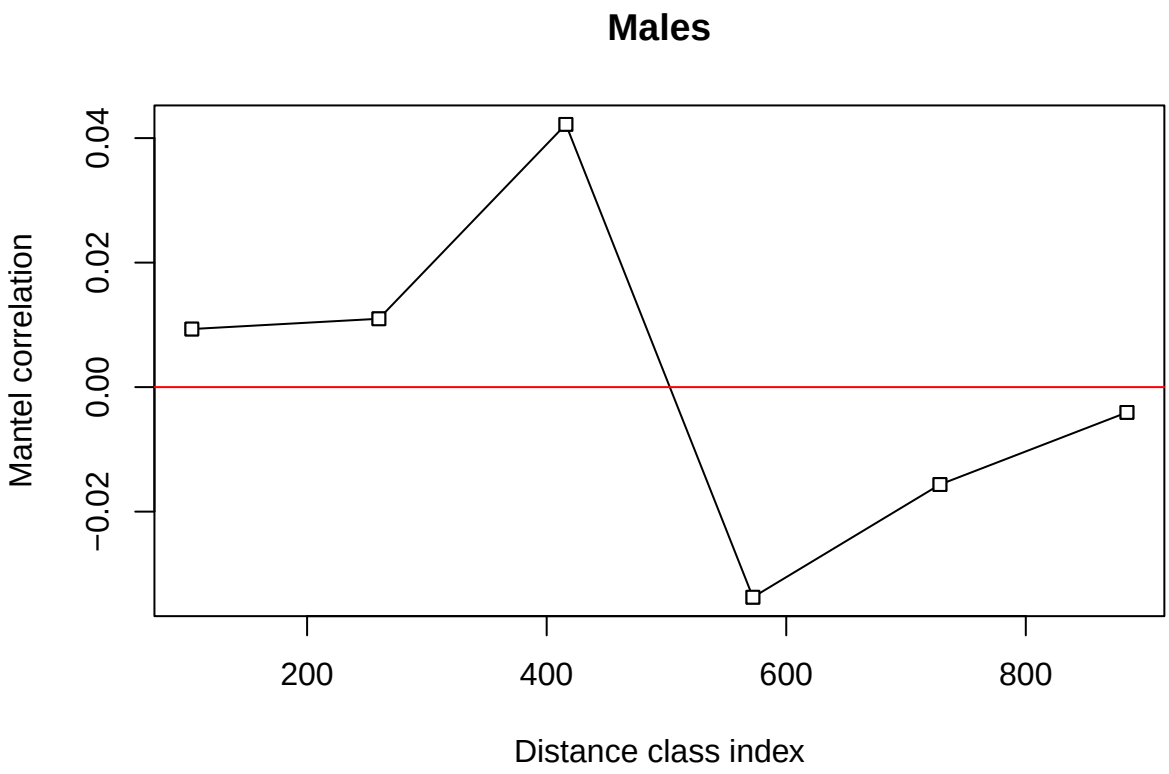
```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = MatrixM, ydis = SpatialMatrixM, permutations = 1000)
##
## Mantel statistic r: 0.03386
##       Significance: 0.31968
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0968 0.1212 0.1523 0.1745
## Permutation: free
## Number of permutations: 1000
```

Mantel correlogram:

```
Mantel_correlog_M <- mantel.correlog(MatrixM, XY=CoordinatesM, nperm=9999)
summary(Mantel_correlog_M)
```

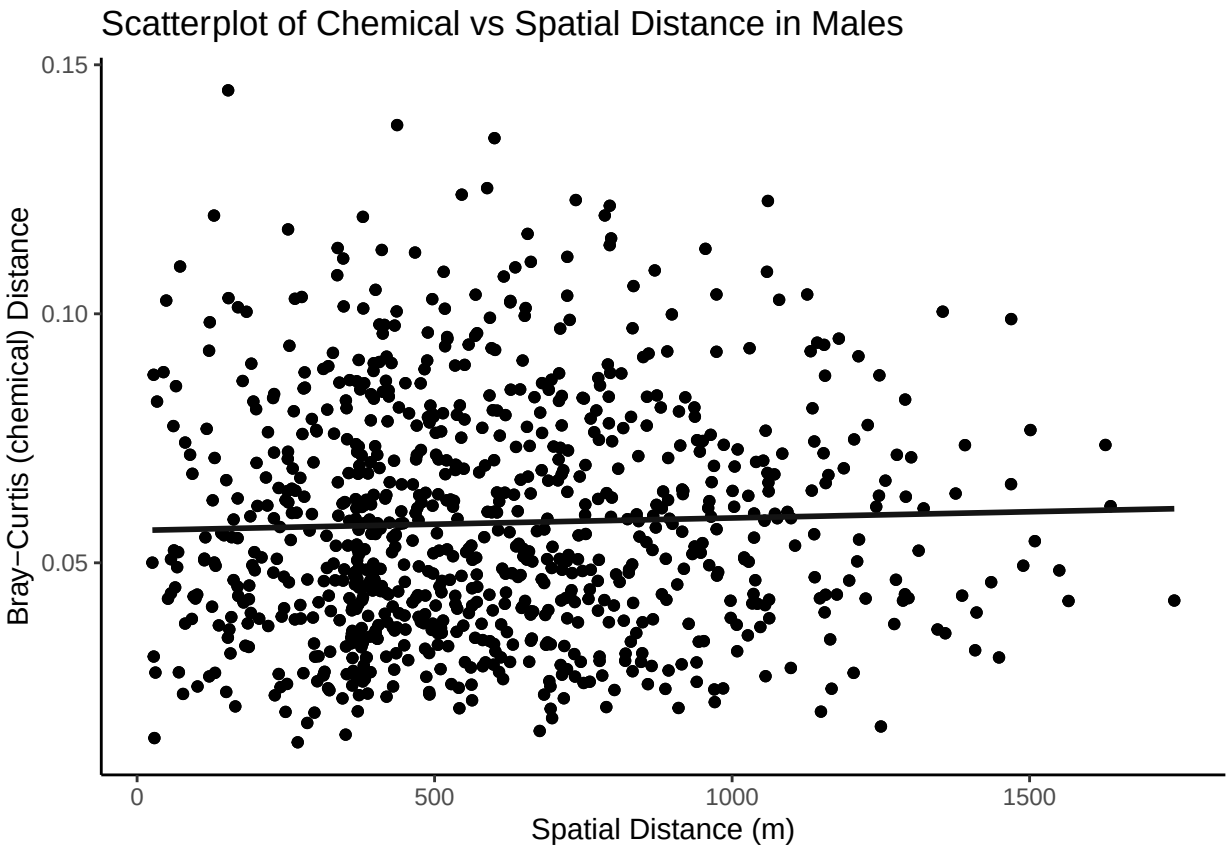
```
##            Length Class  Mode
## mantel.res 55     -none- numeric
## n.class     1     -none- numeric
## break.pts  12     -none- numeric
## mult        1     -none- character
## n.tests     1     -none- numeric
## call        4     -none- call
```

```
MBC <- plot(Mantel_correlog_M)
title(main="Males")
```



Scatterplot:

```
ScatterM <- data.frame(SpatialMatrixM = as.vector(as.matrix(SpatialMatrixM)),
                       MatrixM = as.vector(as.matrix(MatrixM)))%>%
  filter(MatrixM != 0)
ggplot(ScatterM, aes(x=SpatialMatrixM, y=MatrixM))+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "grey8") +
  labs(x = "Spatial Distance (m)", y = "Bray-Curtis (chemical) Distance")+
  ggtitle("Scatterplot of Chemical vs Spatial Distance in Males")
```



4.2 Time difference analysis

A higher similarity between breeding partners could also be the result of a temporal autocorrelation, as breeding partners were sampled at around the same date and time. To control for this, we extracted the pairwise Bray-Curtis similarities of females (N = 50) and males (N = 48) separately, and tested for the effect of the time difference (fixed effect) between each pair of samples on the Bray-Curtis similarity with generalised linear mixed models (GLMM) with beta distribution using the glmmTMB package (Magnusson et al. 2017) in R.

4.2.1 Time difference analysis in females

Preparation of the data

```
# Select samples from all females during nestling-rearing (50 samples)
Females <-Metadata%>%
  filter(Ageclass=="A",
         Breeding_Stage == "N_Rearing",
         Sex == "F",
         Outliers == 0,
         Sample != "L280")%>%
  select(-c(Richness, Shannon_Index, Volatility, pair_brood, f_sampled_twice, Families,
            FamiliesOrdered, Partner_Connected_to_Outlier, F_Connected_to_Outlier, Outliers,
            daysafterhatch, motherID, fatherID))%>%
  mutate(hour_time = (date - 1)*24 + time)%>%
```

```
  arrange(Sample)

# Create a time matrix, containing the time difference in sampling between each female dyad

time_diff <- function(x, y) {
  return(abs(x - y))  # Use "abs()" to get the absolute value of the time difference
}

Time_F <- outer(Females$hour_time, Females$hour_time, FUN = time_diff)
Time_F <- as.matrix(Time_F)
Time_F <- as.data.frame(Time_F)
colnames(Time_F) <- Females$Sample
rownames(Time_F) <- Females$Sample
n <- nrow(Time_F)
for (i in 1:n) { # Remove the upper triangular part of the matrix
  for (j in 1:n) {
    if (i < j) {
      Time_F[i, j] <- NA}}}
for (i in 1:n) { # Remove the values from the main diagonal
  for (j in 1:n) {
    if (i == j) {
      Time_F[i, j] <- 0}}}
```

```
# Create a chemical matrix containing the Bray-Curtis dissimilarity of each female dyad

ChemdataF <- Chemdata%>%
  filter(Sample %in% Females$Sample)%>%
  arrange(Sample)%>% # make sure samples are in the same order as in the Metadata
  select(-Sample) # Select the subset of chemical data

BC_F <- vegdist(ChemdataF) # Calculate Bray-Curtis dissimilarity matrix
BC_F <- as.matrix(BC_F)
BC_F <- as.data.frame(BC_F)
colnames(BC_F) <- Females$Sample
rownames(BC_F) <- Females$Sample
n <- nrow(BC_F)
for (i in 1:n) { # Remove the upper triangular part of the matrix
  for (j in 1:n) {
    if (i < j) {
      BC_F[i, j] <- NA}}}
for (i in 1:n) { # Remove the values from the main diagonal
  for (j in 1:n) {
    if (i == j) {
      BC_F[i, j] <- 0}}}

#Transform Bray-Curtis matrix in right format
BC_F <- as.matrix(BC_F)

key<-data.frame(Sample=Females$Sample, Order=Females$Sample)

all(rownames(BC_F)==key$Sample) # Control if the samples are in the same order in the matrix and in the
```

```
## [1] TRUE
```

Dyadic data

```
# Create a data frame giving for each dyad the time difference in sampling and the Bray-Curtis dissimil

bc_f <- c(as.dist(BC_F))
time_f <- c(as.dist(Time_F))


data.dyad<-data.frame(BC_Dissimilarity=bc_f,
                      Time_Diff=time_f)

list<-expand.grid(key$Sample,key$Sample)
list<-list[which(list$Var1!=list$Var2),]
list$key <- apply(list, 1, function(x)paste(sort(x), collapse=''))
list<-subset(list, !duplicated(list$key))
i=50L # sanity check
BC_F[which(rownames(BC_F)==list$Var1[i]),which(colnames(BC_F)==list$Var2[i])]==bc_f[i]
```

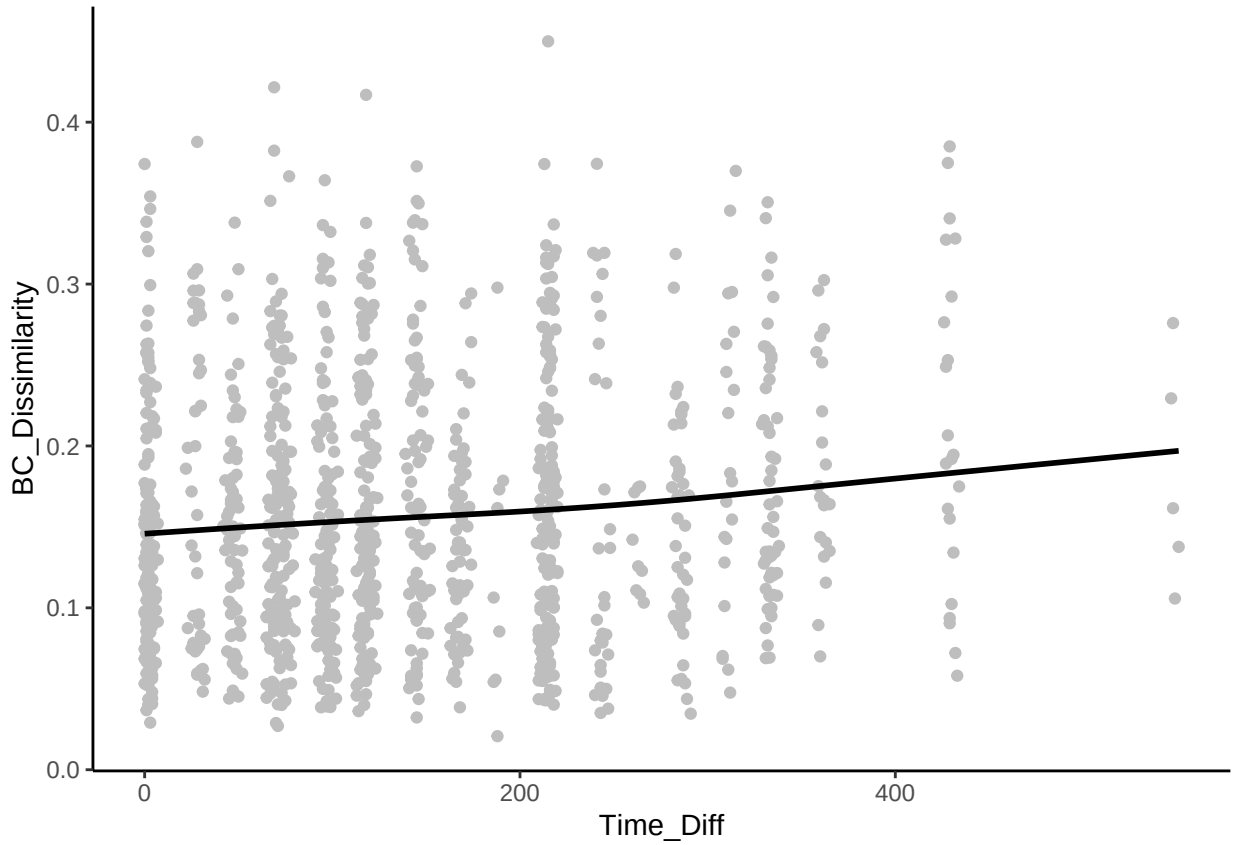
```
## [1] TRUE
```

```
data.dyad$SampleA<-list$Var2
data.dyad$SampleB<-list$Var1

data.dyad<-data.dyad[which(data.dyad$SampleA!=data.dyad$SampleB),] # sanity check

Pairwise_F <- as_tibble(data.dyad)%>%
  dplyr::select("SampleB", everything())%>%
  dplyr::select("SampleA", everything())
```

Analysis and plots

```
ggplot(Pairwise_F)+
  geom_point(aes(x=Time_Diff, y=BC_Dissimilarity), color = "grey")+
  geom_smooth(aes(x=Time_Diff, y=BC_Dissimilarity), color= "black", se=F)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
Model_TimeDiff <- glmmTMB(BC_Dissimilarity ~ Time_Diff,
                          family = beta_family(link = "logit"),
                          data=Pairwise_F)
summary(Model_TimeDiff)
```

```
##  Family: beta  ( logit )
## Formula:          BC_Dissimilarity ~ Time_Diff
## Data: Pairwise_F
##
##      AIC      BIC   logLik deviance df.resid
##  -2941.5  -2926.1   1473.7  -2947.5     1222
##
##
## Dispersion parameter for beta family (): 20.8
##
## Conditional model:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.7688581  0.0270680  -65.35  < 2e-16 ***
## Time_Diff    0.0005815  0.0001486    3.91 9.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(Model_TimeDiff)
```

```
##                     2.5 %        97.5 %      Estimate
```

```
## (Intercept) -1.8219105001 -1.7158057778 -1.7688581390
## Time_Diff    0.0002901767  0.0008728417  0.0005815092
```

The time difference between sampling impacts significantly the chemical similarity of preen oil among females, with a slope of 0.00058

4.2.2 Time difference analysis in males

Preparation of the data

```
# Select samples from all males during nestling-rearing (48 samples)
Males <-Metadata%>%
  filter(Ageclass=="A",
         Breeding_Stage == "N_Rearing",
         Sex == "M",
         Outliers == 0)%>%
  select(-c(Outliers, F_Connected_to_Outlier, Partner_Connected_to_Outlier,
            f_sampled_twice, Families, FamiliesOrdered, clutchsize, broodsize,
            Ageclass, motherID, fatherID, pair_brood, Breeding_Stage))%>%
  mutate(hour_time = (date - 1)*24 + time)%>%
  arrange(Sample)

# Create a time matrix, containing the time difference in sampling between each male dyad

time_diff <- function(x, y) {
  return(abs(x - y))  # Use "abs()" to get the absolute value of the time difference
}

Time_M <- outer(Males$hour_time, Males$hour_time, FUN = time_diff)
Time_M <- as.matrix(Time_M)
Time_M <- as.data.frame(Time_M)
colnames(Time_M) <- Males$Sample
rownames(Time_M) <- Males$Sample
n <- nrow(Time_M)
for (i in 1:n) { # Remove the upper triangular part of the matrix
  for (j in 1:n) {
    if (i < j) {
      Time_M[i, j] <- NA}}}
for (i in 1:n) { # Remove the values from the main diagonal
  for (j in 1:n) {
    if (i == j) {
      Time_M[i, j] <- 0}}}
```

```
# Create a chemical matrix containing the Bray-Curtis dissimilarity of each male dyad

ChemdataM <- Chemdata%>%
  filter(Sample %in% Males$Sample)%>%
  arrange(Sample)%>% # make sure samples are in the same order as in the Metadata
  select(-Sample) # Select the subset of chemical data

BC_M <- vegdist(ChemdataM) # Calculate Bray-Curtis dissimilarity matrix
BC_M <- as.matrix(BC_M)
BC_M <- as.data.frame(BC_M)
```

```
colnames(BC_M) <- Males$Sample
rownames(BC_M) <- Males$Sample
n <- nrow(BC_M)
for (i in 1:n) { # Remove the upper triangular part of the matrix
  for (j in 1:n) {
    if (i < j) {
      BC_M[i, j] <- NA}}}
for (i in 1:n) { # Remove the values from the main diagonal
  for (j in 1:n) {
    if (i == j) {
      BC_M[i, j] <- 0}}}

#Transform Bray-Curtis matrix in right format
BC_M <- as.matrix(BC_M)

key<-data.frame(Sample=Males$Sample, Order=Males$Sample)

all(rownames(BC_M)==key$Sample) # Control if the samples are in the same order in the matrix and Metada
```

```
## [1] TRUE
```

Dyadic data

```
# Create a data frame giving for each dyad the time difference in sampling and the Bray-Curtis dissimil

bc_m <- c(as.dist(BC_M))
time_m <- c(as.dist(Time_M))


data.dyad<-data.frame(BC_Dissimilarity=bc_m,
                      Time_Diff=time_m)

list<-expand.grid(key$Sample,key$Sample)
list<-list[which(list$Var1!=list$Var2),]
list$key <- apply(list, 1, function(x)paste(sort(x), collapse=''))
list<-subset(list, !duplicated(list$key))
i=48L # sanity check
BC_M[which(rownames(BC_M)==list$Var1[i]),which(colnames(BC_M)==list$Var2[i])]==bc_m[i]
```

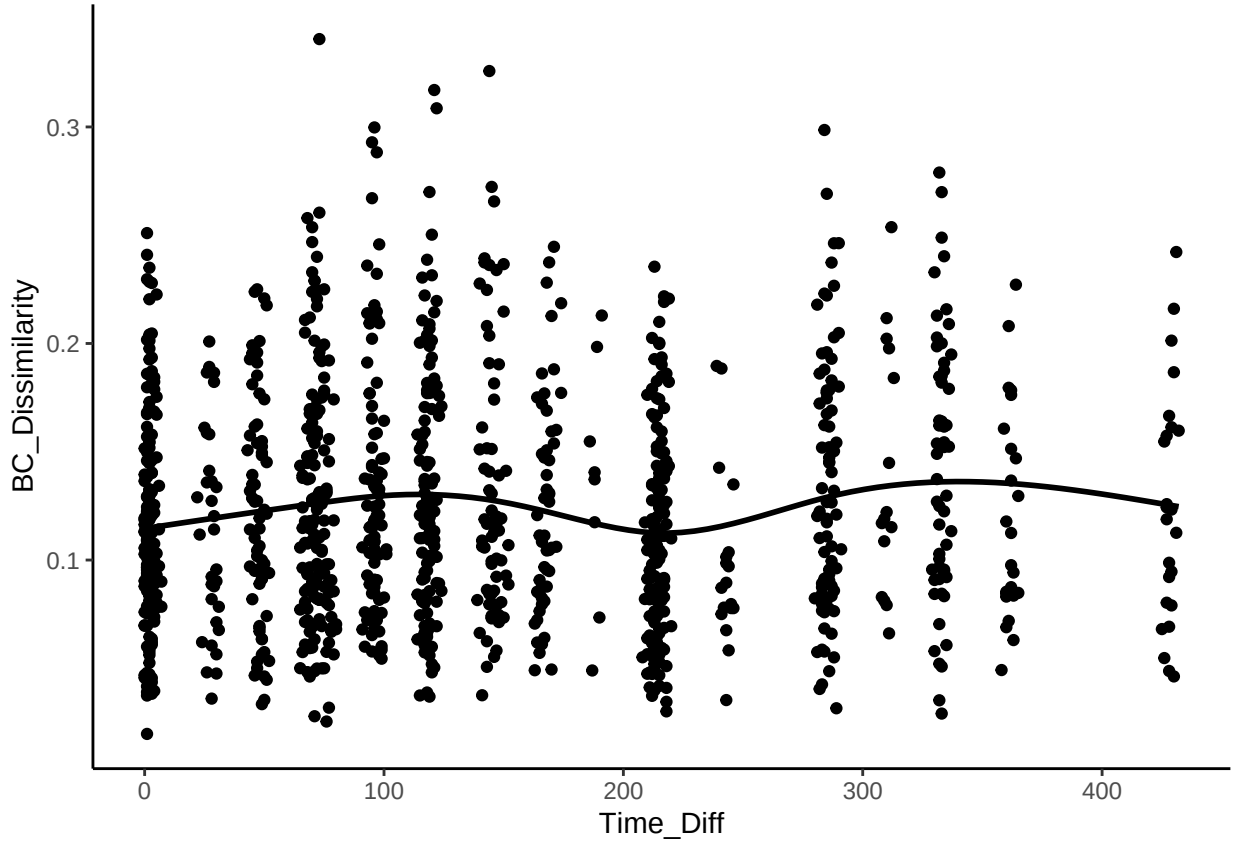
```
## [1] TRUE
```

```
data.dyad$SampleA<-list$Var2
data.dyad$SampleB<-list$Var1

data.dyad<-data.dyad[which(data.dyad$SampleA!=data.dyad$SampleB),] # sanity check

Pairwise_M <- as_tibble(data.dyad)%>%
  dplyr::select("SampleB", everything())%>%
  dplyr::select("SampleA", everything())
```

Analysis and plots

```
ggplot(Pairwise_M)+
  geom_point(aes(x=Time_Diff, y=BC_Dissimilarity))+
  geom_smooth(aes(x=Time_Diff, y=BC_Dissimilarity), color= "black", se=F)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
Model_TimeDiff <- glmmTMB(BC_Dissimilarity ~ Time_Diff,
                          family = beta_family(link = "logit"),
                          data=Pairwise_M)
summary(Model_TimeDiff)
```

```
##  Family: beta  ( logit )
## Formula:          BC_Dissimilarity ~ Time_Diff
## Data: Pairwise_M
##
##      AIC      BIC   logLik deviance df.resid
##  -3488.7  -3473.7   1747.4  -3494.7     1125
##
##
## Dispersion parameter for beta family (): 35.8
##
## Conditional model:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.9853606  0.0238759  -83.15   <2e-16 ***
```

```
## Time_Diff    0.0001841  0.0001292    1.43    0.154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The time difference between sampling does not impact significantly the chemical similarity of preen oil among males

4.3 Life stage difference

We tested whether preen oil composition differs between nestlings and adults, and whether it contains family signatures (i.e. high similarity between family members). We used samples from 16 broods (100 samples, 31 from adults and 69 from nestlings) collected during nestling-rearing, and employed the same analytical method as for the replication analysis. We tested the effect of life stage (fixed effect) and nest identity (random effect) on on beta diversity (Bray-Curtis dissimilarities) using PERMANOVA, PERMDISP (these analyses were ran on PRIMER v7.0.21 and are thus not included in this document) and NMDS, and on chemical richness, Shannon diversity and volatility using LMMs.

```
# Create a subset of the Metadata containing all samples taken from nestboxes for which we have sampled
Families <-Metadata%>%      #100 samples
  filter(Families==1)%>%
  select(-c(Outliers, F_Connected_to_Outlier, Partner_Connected_to_Outlier, f_sampled_twice, pair_brood
```

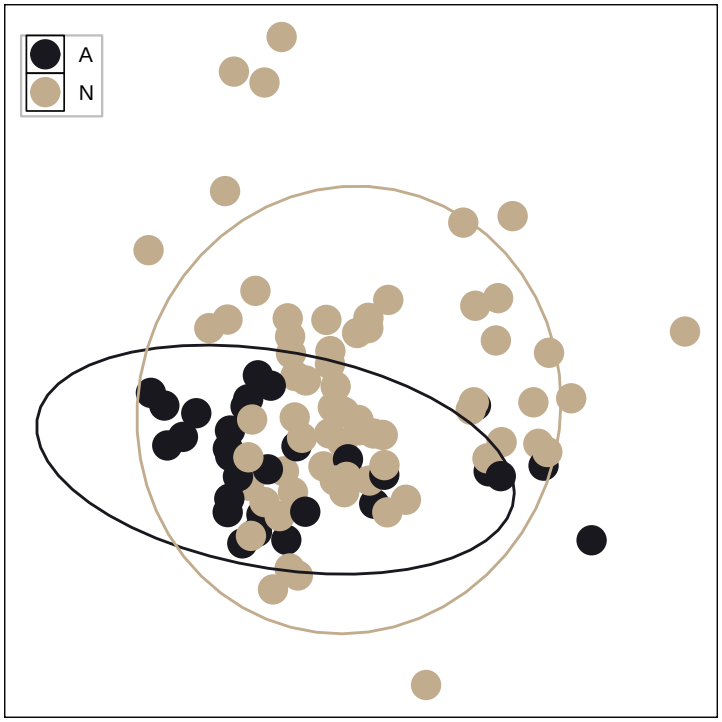
4.3.1 NMDS plot

Bray-Curtis matrix

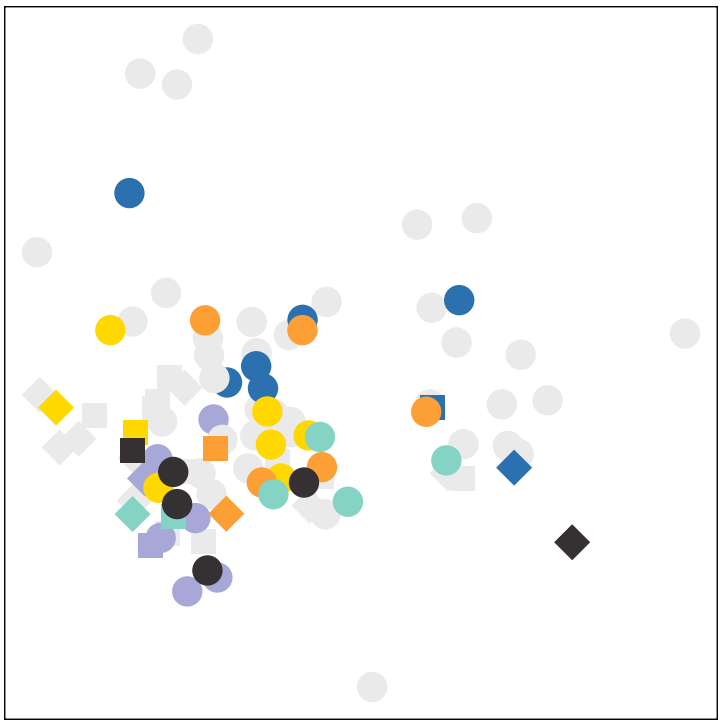
Check the stress (how good the distance between samples in actual multivariate distance is represented in two dimensions)

```
bc$stress
```

```
## [1] 0.1119627
```



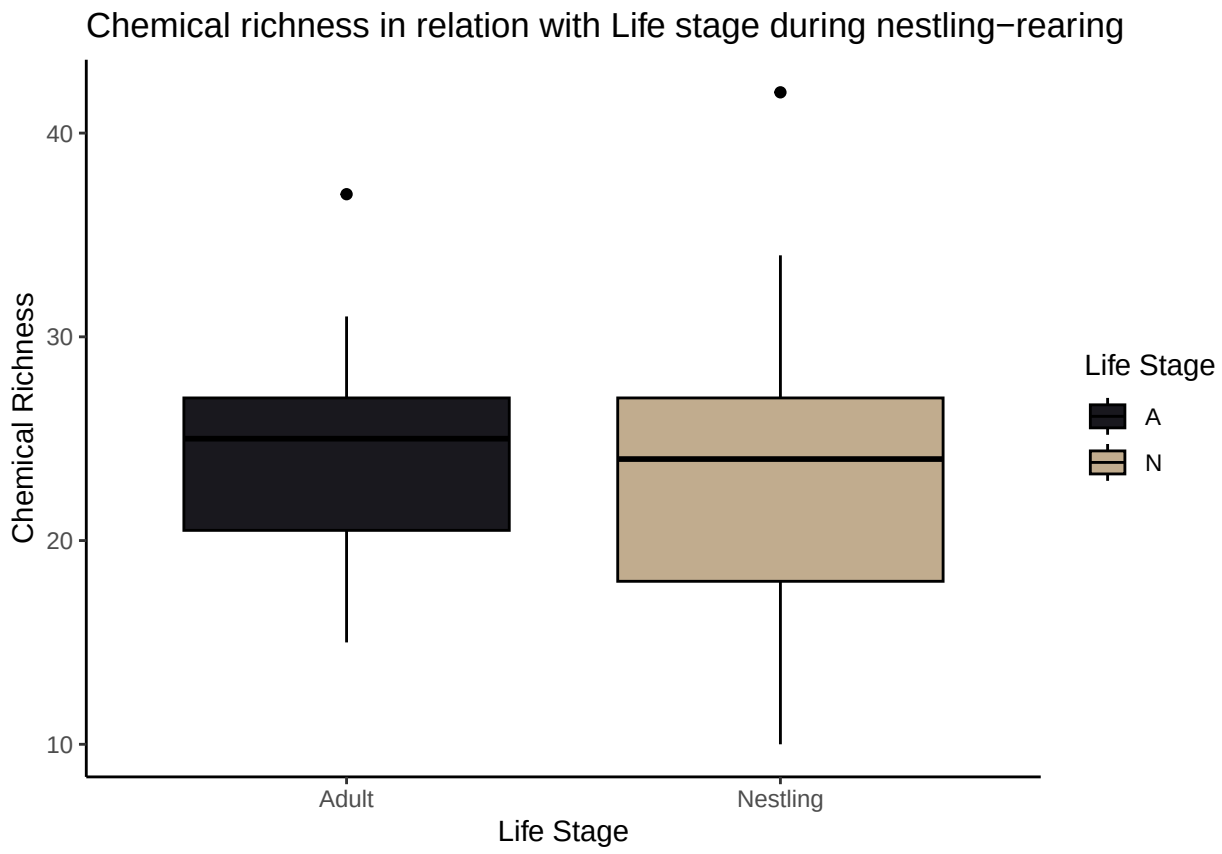
NMDS plot between Life stages



NMDS plot by Family

4.3.2 Richness

a. Visualisation of the life-stage difference (within families) in terms of chemical richness.



b. Analysis

- Fitting the model

```
LMM_LifeStage_R <- lme4::lmer(formula = "Richness ~ Ageclass + (1 | Nestbox_ID)",  data = Families)
summary(LMM_LifeStage_R)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Richness ~ Ageclass + (1 | Nestbox_ID)
##    Data: Families
##
## REML criterion at convergence: 633.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1837 -0.7428  0.1422  0.6684  2.9230
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept)  6.296   2.509
##  Residual               30.741   5.544
## Number of obs: 100, groups:  Nestbox_ID, 16
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)   24.340      1.178  20.656
## AgeclassN     -1.153      1.207  -0.955
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeclassN -0.698
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Richness ~ Ageclass + (1 | Nestbox_ID)",  data = Families))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Richness ~ Ageclass + (1 | Nestbox_ID)"
##    Data: Families
##
## REML criterion at convergence: 633.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1837 -0.7428  0.1422  0.6684  2.9230
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept)  6.296   2.509
##  Residual               30.741   5.544
## Number of obs: 100, groups:  Nestbox_ID, 16
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   24.340      1.178 47.753  20.656   <2e-16 ***
## AgeclassN     -1.153      1.207 84.532  -0.955    0.342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeclassN -0.698
```

P-value = 0.342 –> non significant.

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_LifeStage_R, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term            estimate std.error stati~1 conf.~2 conf.~3
##   <chr>    <chr>      <chr>              <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>       (Intercept)        24.3       1.18  20.7     22.0    26.6
## 2 fixed    <NA>       AgeclassN          -1.15      1.21  -0.955   -3.54    1.42
## 3 ran_pars Nestbox_ID sd__(Intercept)     2.51     NA     NA        0       4.00
## 4 ran_pars Residual   sd__Observation     5.54     NA     NA        4.70    6.38
## # ... with abbreviated variable names 1: statistic, 2: conf.low, 3: conf.high
```

β estimate of sexM effect: -1.15 –> Nestlings have on average 1.15 less substances than adults in our samples. Confidence interval: [-3.33 ; 1.07] –> includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_LifeStage_R, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.0077 0        0.0663   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of Life stage: 0.0077 The Life stage effect only explains 0.77% of the variation of richness in our samples during nestling-rearing.

- Finding the repeatability (random effect)

```
rpt(Richness ~ Ageclass + (1 | Nestbox_ID),
    grname = "Nestbox_ID",
    data = Families, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

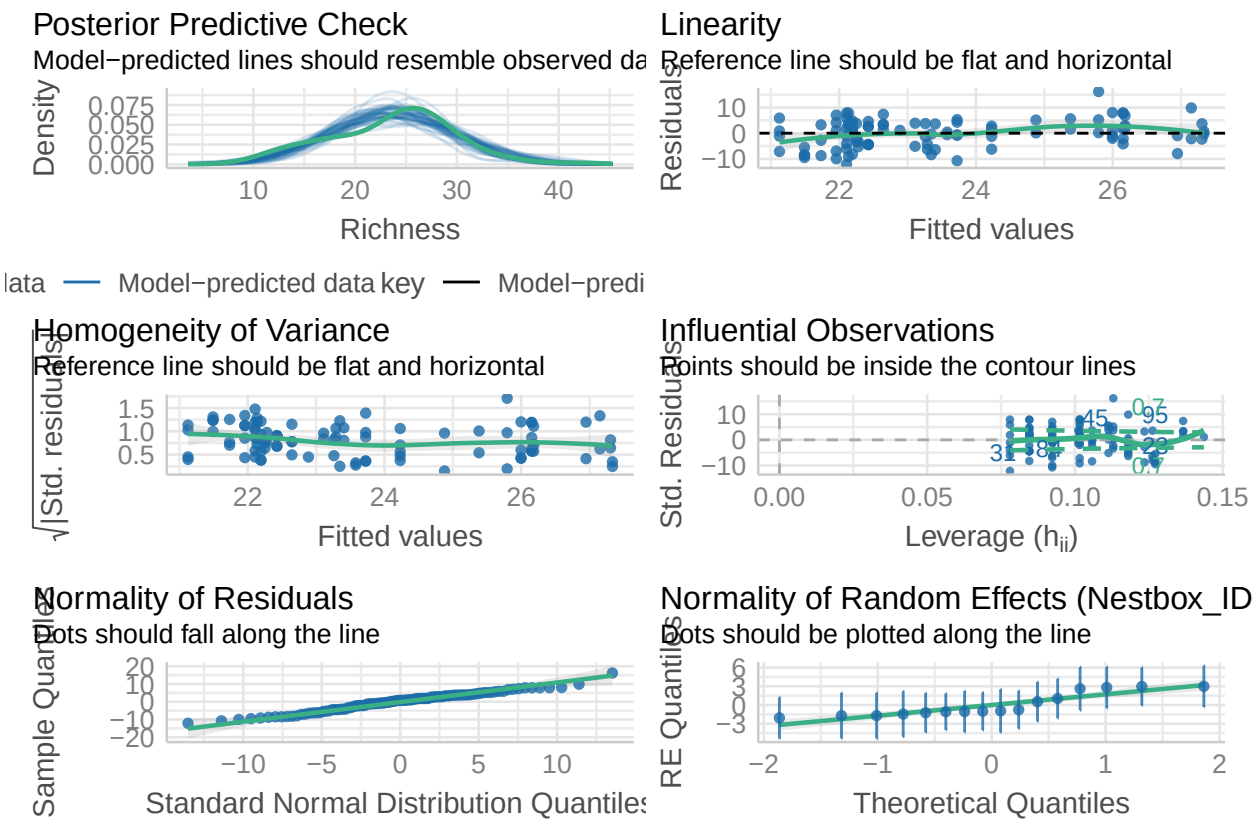
```
## Bootstrap Progress:
## Permutation Progress for Nestbox_ID :
```

```
##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Nestbox_ID
## R  = 0.17
## SE = 0.102
## CI = [0, 0.376]
## P  = 0.0188 [LRT]
##      0.006 [Permutation]
```

Repeatability of the NestboxID effect: 0.17 –> 17% of the variation of richness between samples is due to the variation between families in our data. P(perm)= 0.007, significant. The chemical richness is repeatable between families.

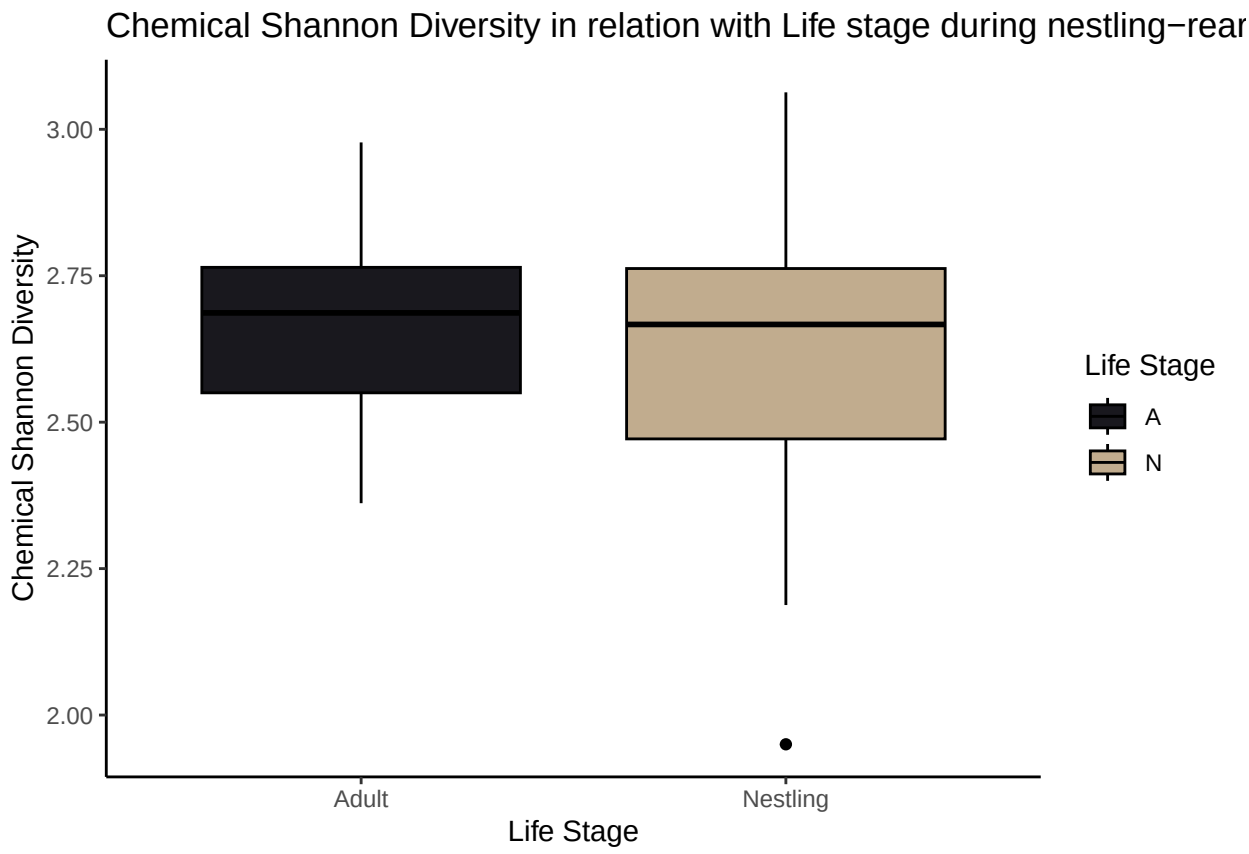
- Model diagnostic

```
check_model(LMM_LifeStage_R)
```



4.3.3 Diversity (Shannon Index)

a. Visualisation of the life-stage difference (within families) in terms of Shannon diversity.



b. Analysis

- Fitting the model

```
LMM_LifeStage_D <- lme4::lmer(formula = "Shannon_Index ~ Ageclass + (1 | Nestbox_ID)",  data = Families)
summary(LMM_LifeStage_D)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Shannon_Index ~ Ageclass + (1 | Nestbox_ID)
##    Data: Families
##
## REML criterion at convergence: -33.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3459 -0.6097  0.2347  0.6453  2.1864
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.003015 0.05491
##  Residual               0.036022 0.18979
## Number of obs: 100, groups:  Nestbox_ID, 16
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)  2.65880    0.03678  72.293
## AgeclassN   -0.04585    0.04121  -1.113
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeclassN -0.767
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Shannon_Index ~ Ageclass + (1 | Nestbox_ID)",  data = Families))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Shannon_Index ~ Ageclass + (1 | Nestbox_ID)"
##    Data: Families
##
## REML criterion at convergence: -33.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3459 -0.6097  0.2347  0.6453  2.1864
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.003015 0.05491
##  Residual               0.036022 0.18979
## Number of obs: 100, groups:  Nestbox_ID, 16
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  2.65880    0.03678 63.46710  72.293   <2e-16 ***
## AgeclassN   -0.04585    0.04121 85.47237  -1.113    0.269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeclassN -0.767
```

P-value = 0.269 –> non significant.

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_LifeStage_D, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term            estimate std.error stati~1 conf.~2 conf.~3
##   <chr>    <chr>      <chr>              <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>       (Intercept)       2.66      0.0368   72.3    2.58   2.73
## 2 fixed    <NA>       AgeclassN        -0.0458    0.0412   -1.11  -0.129  0.0327
## 3 ran_pars Nestbox_ID sd__(Intercept)   0.0549   NA        NA      0      0.0978
## 4 ran_pars Residual   sd__Observation   0.190    NA        NA      0.160  0.218
## # ... with abbreviated variable names 1: statistic, 2: conf.low, 3: conf.high
```

β estimate of Life stage Nestling effect: -0.0458 –> Nestlings have on average a preen oil on average 0.0458 less diverse (Shannon index units) than adults in our samples. Confidence interval: [-0.130 ; 0.0350] –> includes "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_LifeStage_D, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.0115 0        0.0801   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of Life stage: 0.0115 The Life stage effect only explains 1.15% of the variation of diversity in our samples during nestling-rearing.

- Finding the repeatability (random effect)

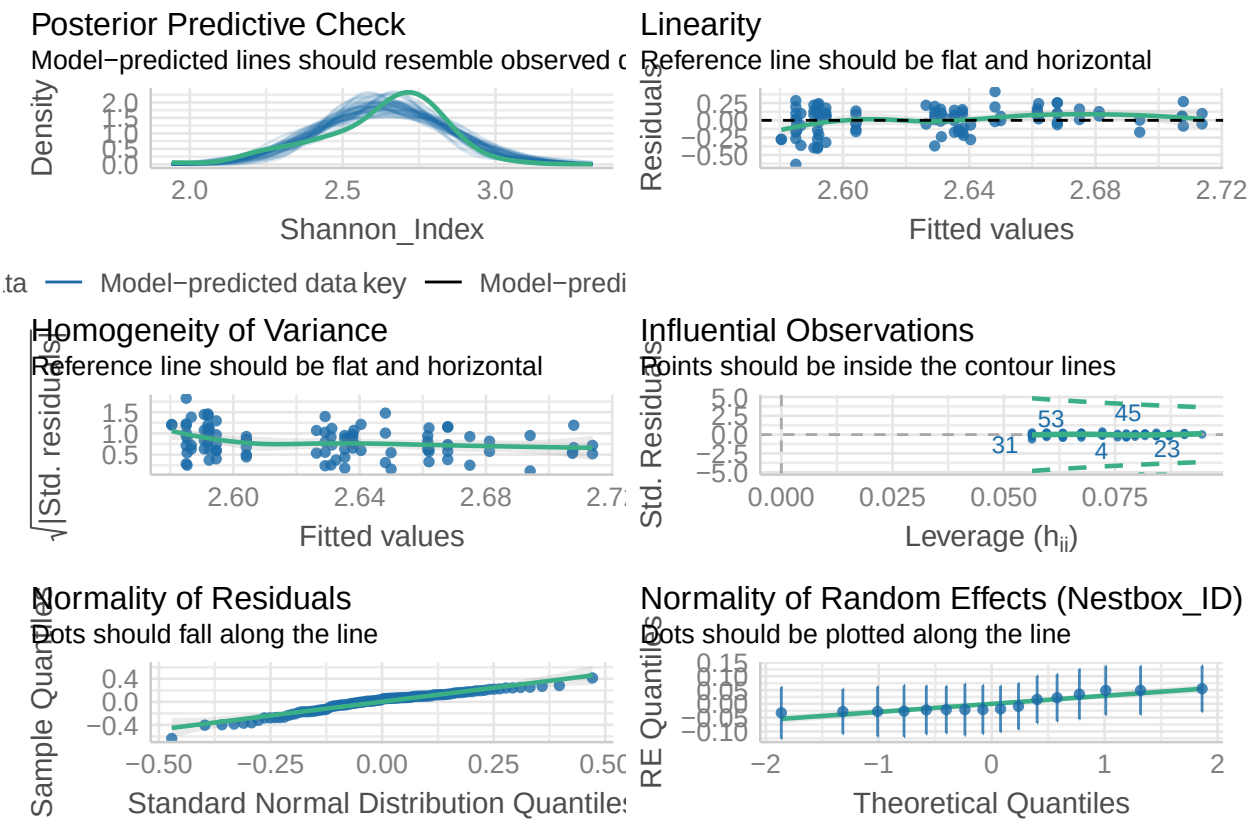
```
rpt(Shannon_Index ~ Ageclass + (1 | Nestbox_ID),
    grname = "Nestbox_ID",
    data = Families, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Nestbox_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Nestbox_ID
## R  = 0.077
## SE = 0.073
## CI = [0, 0.256]
## P  = 0.168 [LRT]
##      0.113 [Permutation]
```

Repeatability of the NestboxID effect: 0.077 –> 7.7% of the variation of diversity between samples is due to the variation between families in our data. P(perm)= 0.1, non-significant. The Shannon diversity is not repeatable between families.

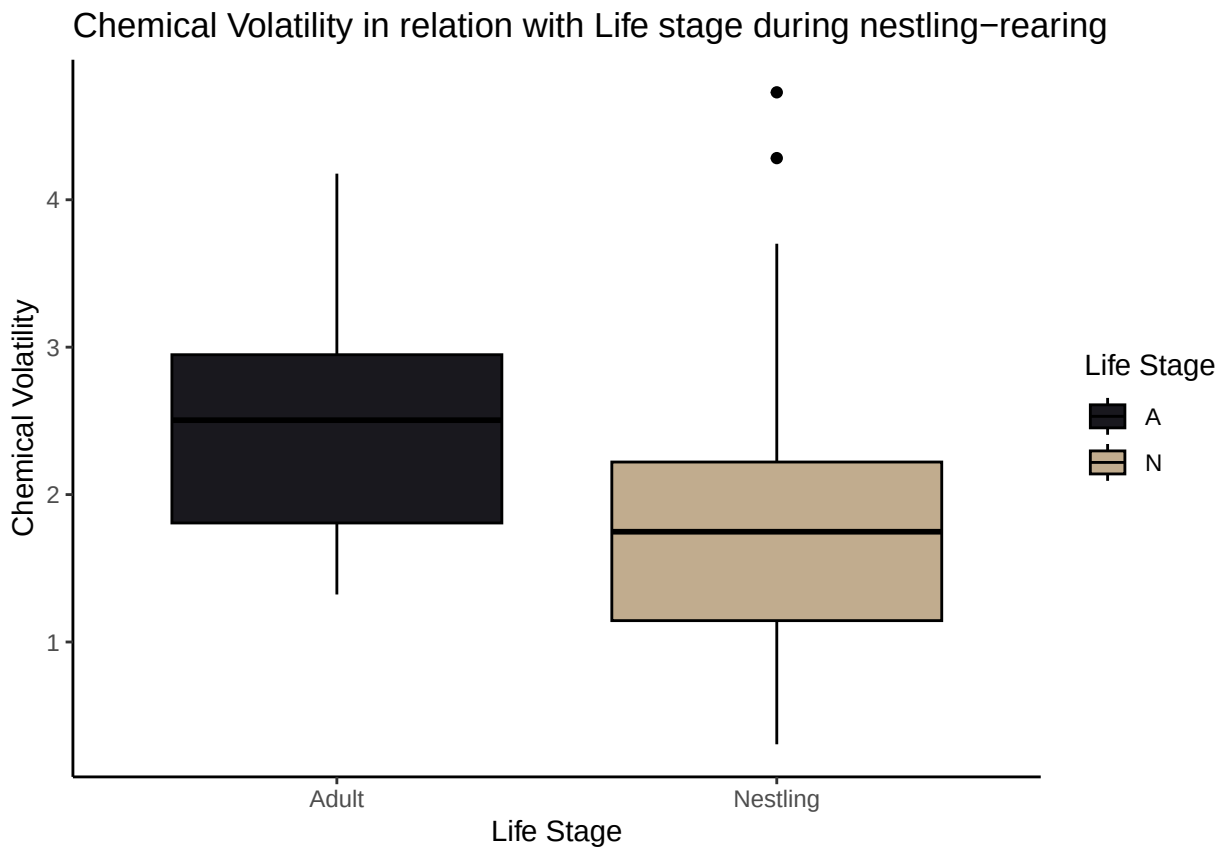
- Model diagnostic

```
check_model(LMM_LifeStage_D)
```



4.3.4 Volatility

a. Visualisation of the life-stage difference (within families) in terms of volatility.



b. Analysis

- Fitting the model

```
LMM_LifeStage_V <- lme4::lmer(formula = "Volatility ~ Ageclass + (1 | Nestbox_ID)",  data = Families)
summary(LMM_LifeStage_V)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Volatility ~ Ageclass + (1 | Nestbox_ID)
##    Data: Families
##
## REML criterion at convergence: 245
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7935 -0.6926 -0.0170  0.4754  3.6392
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.1828   0.4276
##  Residual               0.5576   0.7467
## Number of obs: 100, groups:  Nestbox_ID, 16
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)   2.4443     0.1717  14.234
## AgeclassN    -0.6854     0.1628  -4.209
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeclassN -0.645
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Volatility ~ Ageclass + (1 | Nestbox_ID)",  data = Families))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Volatility ~ Ageclass + (1 | Nestbox_ID)"
##    Data: Families
##
## REML criterion at convergence: 245
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7935 -0.6926 -0.0170  0.4754  3.6392
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Nestbox_ID (Intercept) 0.1828   0.4276
##  Residual               0.5576   0.7467
## Number of obs: 100, groups:  Nestbox_ID, 16
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   2.4443     0.1717 40.8085  14.234  < 2e-16 ***
## AgeclassN    -0.6854     0.1628 84.9299  -4.209 6.34e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## AgeclassN -0.645
```

P-value = 6.34e-05 –> significant.

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_LifeStage_V, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term            estimate std.error stati~1 conf.~2 conf.~3
##   <chr>    <chr>      <chr>              <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>       (Intercept)        2.44      0.172   14.2    2.11    2.77
## 2 fixed    <NA>       AgeclassN         -0.685     0.163   -4.21  -1.00   -0.345
## 3 ran_pars Nestbox_ID sd__(Intercept)    0.428    NA       NA      0.144   0.656
## 4 ran_pars Residual   sd__Observation    0.747    NA       NA      0.624   0.854
## # ... with abbreviated variable names 1: statistic, 2: conf.low, 3: conf.high
```

β estimate of Life stage Nestling effect: -0.007 –> Nestlings have on average a preen oil on average 0.006 less diverse (Shannon index units) than adults in our samples. Confidence interval: [-0.010 ; -0.003] –> does not include "0".

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_LifeStage_V, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.1205 0.032    0.2525   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Marginal R^2 for the effect of Life stage: 0.1205 The Life stage effect explains 12% of the variation of volatility in our samples during nestling-rearing.

- Finding the repeatability (random effect)

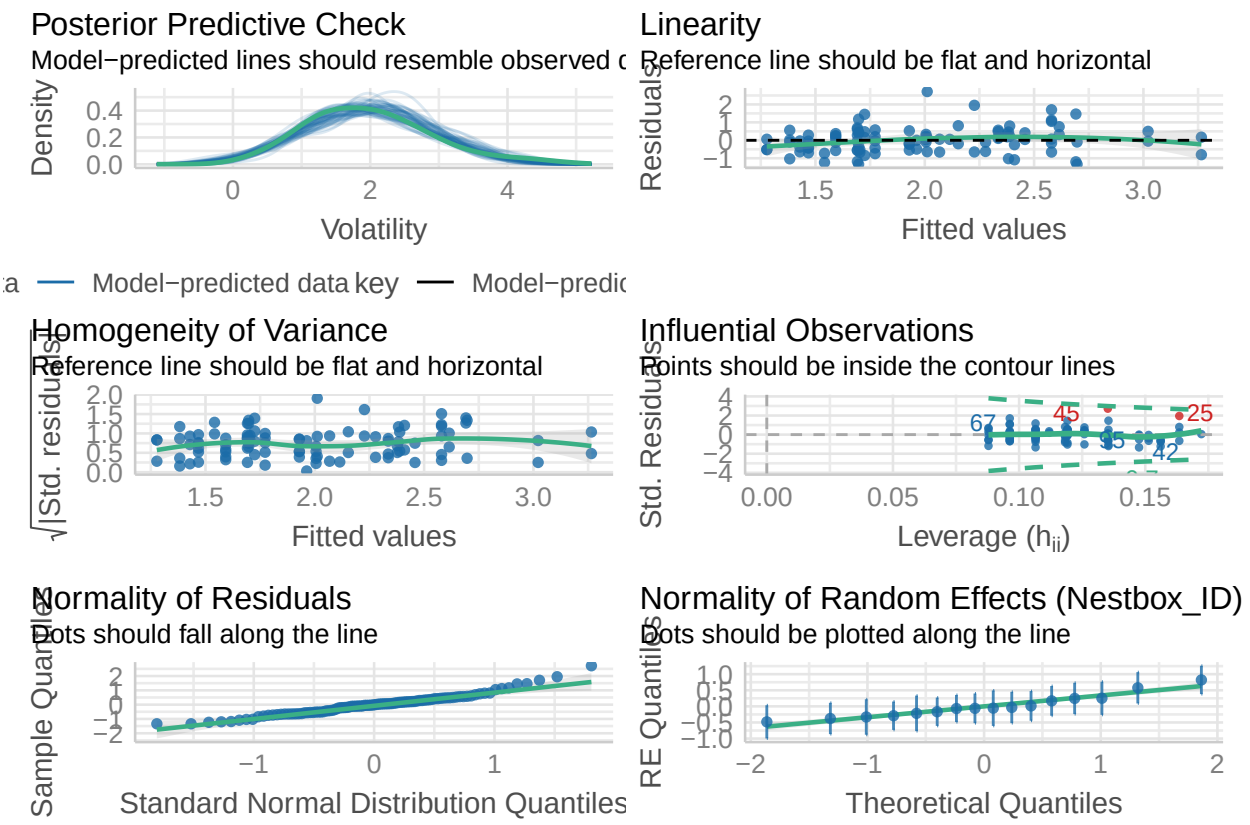
```
rpt(Volatility ~ Ageclass + (1 | Nestbox_ID),
    grname = "Nestbox_ID",
    data = Families, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Nestbox_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Nestbox_ID
## R  = 0.247
## SE = 0.109
## CI = [0.016, 0.445]
## P  = 0.000773 [LRT]
##      0.002 [Permutation]
```

Repeatability of the NestboxID effect: 0.247 –> 24.7% of the variation of volatility between samples is due to the variation between families in our data. P(perm)= 0.002, significant. The Volatility is repeatable between families.

- Model diagnostic

```
check_model(LMM_LifeStage_V)
```



4.4 Nestling-Adult similarities

We tested whether the preen oil from nestlings is more similar to that of their mother or father, to that of an adult female or male other than their mother and father, and whether it was more similar to that of their mother or father more so than to other adult females or males in the population. First, we extracted the pairwise Bray-Curtis similarity for each nestling-adult pair from the Bray-Curtis matrix, and separated them between nestling-mother, nestling-father, nestling-adult female and nestling-adult male pairs. We could then study the effect of adult and parent sex, as well as the effect of being the mother/father (fixed effect) on the similarity between samples, while controlling for the effect of nest identity (random effect) and nestling identity (random effect nested within nest identity). As Bray-Curtis similarity data ranges between 0 and 1, we decided to use generalised linear mixed models (GLMM) with Beta distribution using the *glmmTMB* package (Magnusson et al. 2017) on R. However this method does not allow us to measure the repeatability for random effects. Instead, to estimate the importance of random effects, we ran models with and without each random effect and compared them with a chi-square test using the *stats* package.

4.4.1 Creating a Pairwise-Similarity data file

This code follows the code from Raulo et al. (2021) (full reference in the references of the main paper)

```
#### Create Bray-Curtis matrix with the Families samples
```

```r
# Recreate the subset of data containing the 100 family samples ordered
Metadata_Families <- Metadata%>%
  filter(Families == 1,
         Outliers == 0)%>%
  select(c(Sample, Individual_ID, Nestbox_ID,
  Ageclass, Sex, Families, FamiliesOrdered))%>%
  rename(Order = FamiliesOrdered)

ChemdataF <- Chemdata%>%
  filter(Sample %in% Metadata_Families$Sample)%>%
  select(-Sample) # Select the subset of chemical data

BC_Families <- vegdist(ChemdataF)
BC_Families <- as.matrix(BC_Families)
BC_Families <- 1-BC_Families # to have bray-curtis similarity instead of dissimilarity
BC_Families <- as.data.frame(BC_Families)
colnames(BC_Families) <- Metadata_Families$Sample
rownames(BC_Families) <- Metadata_Families$Sample
n <- nrow(BC_Families)
for (i in 1:n) { # Remove the upper triangular part of the matrix
  for (j in 1:n) {
    if (i < j) {
      BC_Families[i, j] <- NA}}}
for (i in 1:n) { # Remove the values from the main diagonal
  for (j in 1:n) {
    if (i == j) {
      BC_Families[i, j] <- 0}}}

#Transform Bray-Curtis in right format
BC_Families <- as.matrix(BC_Families)

key<-data.frame(Sample=Metadata_Families$Sample, Order=Metadata_Families$Order)

all(rownames(BC_Families)==key$Sample) # Control if the samples are in the same order in the matrix and
```

```
## [1] TRUE
```

```r
####Create a binary matrix, value = 1 when the samples come from the same nestbox, 0 otherwise

# 1. Create data frame  with each Individual name (SampleID) and their nestbox (NestID) as character (h
NestID_frame <- Metadata_Families[,c("Sample", "Nestbox_ID")]

# 2. Create an empty numeric matrix to fill with distances
NestM <- array(0,c(nrow(NestID_frame),nrow(NestID_frame)))

# 3. Derive matrix with binary NestID similarity between each sample (they are either from the same nes
for(i in 1:nrow(NestID_frame)){
  for(j in 1:nrow(NestID_frame)){
    if(NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestM[i,j]= 1
    } else{
      NestM[i,j]= 0
    }
```

```
  }
}
all(rownames(NestM)==key$Sample)
```

```
## [1] TRUE
```

```
rownames(NestM)<-key$Sample
colnames(NestM)<-key$Sample
```

```
#### Create a matrix with the BC distances
# 1. Create an empty numeric matrix to fill with nestID
NestIDM <-array(as.character(NA),c(nrow(NestID_frame),nrow(NestID_frame)))

# 2. Derive matrix with binary NestID similarity between each sample
for(i in 1:nrow(NestID_frame)){
  for(j in 1:nrow(NestID_frame)){
    if(NestID_frame$Nestbox_ID[i]=="15" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "15"}
    if(NestID_frame$Nestbox_ID[i]=="26" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "26"}
    if(NestID_frame$Nestbox_ID[i]=="121" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "121"}
    if(NestID_frame$Nestbox_ID[i]=="151" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "151"}
    if(NestID_frame$Nestbox_ID[i]=="222" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "222"}
    if(NestID_frame$Nestbox_ID[i]=="301" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "301"}
    if(NestID_frame$Nestbox_ID[i]=="308" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "308"}
    if(NestID_frame$Nestbox_ID[i]=="447" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "447"}
    if(NestID_frame$Nestbox_ID[i]=="512" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "512"}
    if(NestID_frame$Nestbox_ID[i]=="715" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "715"}
    if(NestID_frame$Nestbox_ID[i]=="39L" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "39L"}
    if(NestID_frame$Nestbox_ID[i]=="519A" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "519A"}
    if(NestID_frame$Nestbox_ID[i]=="E" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "E"}
    if(NestID_frame$Nestbox_ID[i]=="M" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "M"}
    if(NestID_frame$Nestbox_ID[i]=="O" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "O"}
    if(NestID_frame$Nestbox_ID[i]=="Y" & NestID_frame$Nestbox_ID[i]==NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "Y"}
    if(NestID_frame$Nestbox_ID[i]!=NestID_frame$Nestbox_ID[j]){
      NestIDM[i,j]= "NA"}
  }
}
```

```
all(rownames(NestIDM)==key$Sample)
```

```
## [1] TRUE
```

```
rownames(NestIDM)<-key$Sample
colnames(NestIDM)<-key$Sample
```

```
####Create a combination-factor matrix for Life Stage (Adult vs Nestling)

# 1. Create data frame  with each Individual name (SampleID) and their Life stage (ageclass) as charact
LifeStage_frame <- Metadata_Families[,c("Sample", "Ageclass")]

# 2. Create an empty character matrix to fill with characters
AGEM<-array(as.character(NA),c(nrow(LifeStage_frame),nrow(LifeStage_frame)))

for(i in 1:nrow(LifeStage_frame)){
  for(j in 1:nrow(LifeStage_frame)){
    if(LifeStage_frame$Ageclass[i]=="A" & LifeStage_frame$Ageclass[i]==LifeStage_frame$Ageclass[j]){
      AGEM[i,j]= "AA"}
    if(LifeStage_frame$Ageclass[i]=="N" & LifeStage_frame$Ageclass[i]==LifeStage_frame$Ageclass[j]){
      AGEM[i,j]= "NN"}
    if( LifeStage_frame$Ageclass[i]!=LifeStage_frame$Ageclass[j]){
      AGEM[i,j]= "AN"}
  }
}

rownames(AGEM)<-key$Sample
colnames(AGEM)<-key$Sample
all(rownames(AGEM)==key$Sample)
```

```
## [1] TRUE
```

```
####Create a combination-factor matrix for Sexes

Sex_frame<-Metadata_Families[,c("Sample","Sex")]
SEXM<-array(as.character(NA),c(nrow(Sex_frame),nrow(Sex_frame)))

for(i in 1:nrow(Sex_frame)){
  for(j in 1:nrow(Sex_frame)){
    if(Sex_frame$Sex[i]=="F" & Sex_frame$Sex[i]==Sex_frame$Sex[j]){
      SEXM[i,j]= "FF"}
    if(Sex_frame$Sex[i]=="M" & Sex_frame$Sex[i]==Sex_frame$Sex[j]){
      SEXM[i,j]= "MM"}
    if(Sex_frame$Sex[i]=="NA" & Sex_frame$Sex[i]==Sex_frame$Sex[j]){
      SEXM[i,j]= "NN"}
    if(Sex_frame$Sex[i]=="F" & Sex_frame$Sex[j]=="M"){
      SEXM[i,j]= "FM"}
    if(Sex_frame$Sex[j]=="F" & Sex_frame$Sex[i]=="M"){
      SEXM[i,j]= "FM"}
    if(Sex_frame$Sex[i]=="F" & Sex_frame$Sex[j]=="NA"){
      SEXM[i,j]= "NF"}
```

```
    if(Sex_frame$Sex[j]=="F" & Sex_frame$Sex[i]=="NA"){
      SEXM[i,j]= "NF"}
    if(Sex_frame$Sex[i]=="M" & Sex_frame$Sex[j]=="NA"){
      SEXM[i,j]= "NM"}
    if(Sex_frame$Sex[j]=="M" & Sex_frame$Sex[i]=="NA"){
      SEXM[i,j]= "NM"}
  }
}
rownames(SEXM)<-key$Sample
colnames(SEXM)<-key$Sample
```

```
#### Create dyadic data

bc_families <- c(as.dist(BC_Families))
nest <- c(as.dist(NestM))
age <- c(AGEM[lower.tri(AGEM)])
sex<-c(SEXM[lower.tri(SEXM)])
nestID<-c(NestIDM[lower.tri(SEXM)])

data.dyad<-data.frame(BC_Similarity=bc_families,
                      Nest_Similarity=nest,
                      NestID=nestID,
                      Age_combination=age,
                      Sex_combination=sex)

list<-expand.grid(key$Sample,key$Sample)
list<-list[which(list$Var1!=list$Var2),]
list$key <- apply(list, 1, function(x)paste(sort(x), collapse=''))
list<-subset(list, !duplicated(list$key))
i=90 # sanity check
BC_Families[which(rownames(BC_Families)==list$Var1[i]),which(colnames(BC_Families)==list$Var2[i])]==bc_
```

```
## [1] TRUE
```

```
data.dyad$SampleA<-list$Var2
data.dyad$SampleB<-list$Var1

data.dyad<-data.dyad[which(data.dyad$SampleA!=data.dyad$SampleB),] # sanity check

Pairwise_Data <- as_tibble(data.dyad)%>%
  dplyr::select("SampleB", everything())%>%
  dplyr::select("SampleA", everything())
```

4.4.2 Mother VS father analysis

```
Parent_Data <- Pairwise_Data%>%
  mutate(Relationship = case_when(
    Sex_combination == 'NM' & Nest_Similarity == '1' ~ 'Father',
    Sex_combination == 'NF' & Nest_Similarity == '1' ~ 'Mother'
  ))%>%
```

```
  filter(Relationship != "NA")
Age_of_Samples <- Metadata_Families%>%
  select(Sample, Ageclass)%>%
  rename(SampleA = Sample)
df_list <- list(Parent_Data, Age_of_Samples)
Parent_Data <- df_list%>%
  reduce(full_join, by="SampleA")%>%
  filter(SampleB != "NA")%>%
  mutate(SampleB = as.character(as.factor(SampleB)),
    Nestling = ifelse(Ageclass == 'A', SampleB, SampleA))%>%
  select(-c(Ageclass, Nest_Similarity, Age_combination, Sex_combination))
```

4.4.2.1 Analyse

- Fittig model and getting the P-value

```
Model_Parent_Nestling <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling),
                                 family = beta_family(link = "logit"),
                                 data=Parent_Data)
summary(Model_Parent_Nestling) #AIC = -380.8
```

```
##  Family: beta  ( logit )
## Formula:
## BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling)
## Data: Parent_Data
##
##      AIC      BIC   logLik deviance df.resid
##   -380.8   -366.3    195.4   -390.8      130
##
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  NestID          (Intercept) 0.10333  0.3215
##  NestID:Nestling (Intercept) 0.07092  0.2663
## Number of obs: 135, groups:  NestID, 16; NestID:Nestling, 69
##
## Dispersion parameter for beta family (): 68.4
##
## Conditional model:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.72361    0.09687  17.794   <2e-16 ***
## RelationshipMother -0.12897    0.05694  -2.265   0.0235 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-value = 0.0235, significant. Mother's preen oil composition is slightly more distant to their nestlings preen oil than the father's preen oil.

```
#Conversion in linear unit
mu_intercept <- 1.72361
mu_ParentMother <- -0.12897
plogis(mu_intercept + mu_ParentMother) - plogis(mu_intercept)
```

```
## [1] -0.01732533
```

The distance Mother-Nestling is 0.017 higher than the distance Father-Nestling in Bray-Curtis units.

- Confidence intervals

```
tidy(Model_Parent_Nestling, conf.int = TRUE, conf.method = "profile")
```

```
## # A tibble: 4 x 10
##   effect   compo~1 group term  estim~2 std.e~3 stati~4   p.value conf.~5 conf.~6
##   <chr>    <chr>   <chr> <chr>   <dbl>   <dbl>   <dbl>     <dbl>   <dbl>   <dbl>
## 1 fixed    cond    <NA>  (Int~   1.72   0.0969   17.8   7.87e-71   1.52   1.92
## 2 fixed    cond    <NA>  Rela~  -0.129  0.0569   -2.27  2.35e- 2  -0.243 -0.0163
## 3 ran_pars cond    Nest~ sd__~   0.321 NA        NA     NA         -1.66  -0.649
## 4 ran_pars cond    Nest~ sd__~   0.266 NA        NA     NA         -1.79  -0.995
## # ... with abbreviated variable names 1: component, 2: estimate, 3: std.error,
## #   4: statistic, 5: conf.low, 6: conf.high
```

β estimate of the distance to the Mother effect: -0.129 (Beta family unit) Confidence interval: [-0.242 ; -0.016] –> does not include “0”.

- AIC without Random effects

```
Model_PN_noRandomeff <- glmmTMB(BC_Similarity ~ Relationship,
                                family = beta_family(link = "logit"),
                                data=Parent_Data)

Model_PN_NestID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID),
                                 family = beta_family(link = "logit"),
                                 data=Parent_Data)

Model_PN_NestlingID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID:Nestling),
                               family = beta_family(link = "logit"),
                               data=Parent_Data)

anova(Model_Parent_Nestling, Model_PN_noRandomeff, Model_PN_NestID, Model_PN_NestlingID)
```

```
## Data: Parent_Data
## Models:
## Model_PN_noRandomeff: BC_Similarity ~ Relationship, zi=~0, disp=~1
## Model_PN_NestID: BC_Similarity ~ Relationship + (1 | NestID), zi=~0, disp=~1
## Model_PN_NestlingID: BC_Similarity ~ Relationship + (1 | NestID:Nestling), zi=~0, disp=~1
## Model_Parent_Nestling: BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling), zi=~0, di
##                        Df     AIC     BIC logLik deviance  Chisq Chi Df
## Model_PN_noRandomeff    3 -341.38 -332.66 173.69  -347.38
## Model_PN_NestID         4 -371.10 -359.48 189.55  -379.10 31.721      1
## Model_PN_NestlingID     4 -367.99 -356.37 188.00  -375.99  0.000      0
## Model_Parent_Nestling   5 -380.78 -366.26 195.39  -390.78 14.791      1
##                        Pr(>Chisq)
## Model_PN_noRandomeff
## Model_PN_NestID          1.78e-08 ***
```

```
## Model_PN_NestlingID    1.0000000
## Model_Parent_Nestling  0.0001201 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The stronger AIC is for the full model. There is a significant difference between the model without random effects and the model without the NestlingID effect, but no with the model without NestID effect. The NestlingID effect seems to be the random effect with the most impact.

4.4.3 Mother VS other adult females analysis

```
MotherVSOtherF_Data <- Pairwise_Data%>%
  mutate(Relationship = case_when(
    Sex_combination == 'NF' & Nest_Similarity == '0' ~ 'OtherF',
    Sex_combination == 'NF' & Nest_Similarity == '1' ~ 'Mother'
  ))%>%
  filter(Relationship != "NA")
Age_of_Samples <- Metadata_Families%>%
  select(Sample, Ageclass)%>%
  rename(SampleA = Sample)
df_list <- list(MotherVSOtherF_Data, Age_of_Samples)
MotherVSOtherF_Data <- df_list%>%
  reduce(full_join, by="SampleA")%>%
  filter(SampleB != "NA")%>%
  mutate(SampleB = as.character(as.factor(SampleB)),
    Nestling = ifelse(Ageclass == 'A', SampleB, SampleA))%>%
  select(-c(Ageclass, Nest_Similarity, Age_combination, Sex_combination))
```

4.4.3.1 Analyse

- Fittig model and getting the P-value

```
Model_Female_Nestling <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling),
                                 family = beta_family(link = "logit"),
                                 data=MotherVSOtherF_Data)
summary(Model_Female_Nestling) #AIC = -6970.3
```

```
##  Family: beta  ( logit )
## Formula:
## BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling)
## Data: MotherVSOtherF_Data
##
##      AIC      BIC   logLik deviance df.resid
##  -3093.3  -3068.6   1551.7  -3103.3     1030
##
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  NestID          (Intercept) 0.07611  0.2759
```

```
##  NestID:Nestling (Intercept) 0.09286  0.3047
## Number of obs: 1035, groups:  NestID, 16; NestID:Nestling, 135
##
## Dispersion parameter for beta family (): 56.3
##
## Conditional model:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.58318    0.09296   17.03   <2e-16 ***
## RelationshipOtherF -0.01167    0.29364   -0.04    0.968
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-value = 0.968, non-significant. Mother's preen oil composition is not more similar to that of their nestlings than that of another female in the population.

```
#Conversion in linear unit
mu_intercept <- 1.58318
mu_OtherF <- -0.01167
plogis(mu_intercept + mu_OtherF) - plogis(mu_intercept)
```

```
## [1] -0.001655648
```

The distance Mother-Nestling is 0.002 higher than the distance OtherFemale-Nestling in Bray-Curtis units.

- Confidence intervals

```
confint(Model_Female_Nestling)
```

```
##                                       2.5 %    97.5 %    Estimate
## (Intercept)                        1.4009823 1.7653853  1.58318381
## RelationshipOtherF                -0.5871983 0.5638497 -0.01167433
## Std.Dev.(Intercept)|NestID         0.1504672 0.5058573  0.27588932
## Std.Dev.(Intercept)|NestID:Nestling  0.2596813 0.3576063  0.30473540
```

β estimate of the distance to adult females effect: -0.0016 (Beta family unit) Confidence interval: [-0.587 ; 0.564] –> includes "0".

- AIC without Random effects

```
Model_FN_noRandomeff <- glmmTMB(BC_Similarity ~ Relationship,
                                family = beta_family(link = "logit"),
                                data=MotherVSOtherF_Data)
Model_FN_noNestlingID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID),
                                 family = beta_family(link = "logit"),
                                 data=MotherVSOtherF_Data)
Model_FN_noNestID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID:Nestling),
                             family = beta_family(link = "logit"),
                             data=MotherVSOtherF_Data)
anova(Model_Female_Nestling, Model_FN_noRandomeff, Model_FN_noNestlingID, Model_FN_noNestID)
```

```
## Data: MotherVSOtherF_Data
## Models:
## Model_FN_noRandomeff: BC_Similarity ~ Relationship, zi=~0, disp=~1
## Model_FN_noNestlingID: BC_Similarity ~ Relationship + (1 | NestID), zi=~0, disp=~1
## Model_FN_noNestID: BC_Similarity ~ Relationship + (1 | NestID:Nestling), zi=~0, disp=~1
## Model_Female_Nestling: BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling), zi=~0, di
##                        Df     AIC     BIC logLik deviance    Chisq Chi Df
## Model_FN_noRandomeff    3 -2653.3 -2638.5 1329.7  -2659.3
## Model_FN_noNestlingID   4 -2654.9 -2635.1 1331.5  -2662.9   3.5827      1
## Model_FN_noNestID       4 -3088.7 -3068.9 1548.3  -3096.7 433.8058      0
## Model_Female_Nestling   5 -3093.3 -3068.6 1551.7  -3103.3   6.6411      1
##                        Pr(>Chisq)
## Model_FN_noRandomeff
## Model_FN_noNestlingID    0.058384 .
## Model_FN_noNestID       < 2.2e-16 ***
## Model_Female_Nestling    0.009965 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The stronger AIC is for the full model. There is a significant difference between the model without random effects and the model without the NestID effect, but no with the model without NestlingID effect. The NestID effect seems to be the random effect with the most impact.

4.4.4 Father VS other adult males analysis

```
FatherVSOtherM_Data <- Pairwise_Data%>%
  mutate(Relationship = case_when(
    Sex_combination == 'NM' & Nest_Similarity == '0' ~ 'OtherM',
    Sex_combination == 'NM' & Nest_Similarity == '1' ~ 'Father'
  ))%>%
  filter(Relationship != "NA")
Age_of_Samples <- Metadata_Families%>%
  select(Sample, Ageclass)%>%
  rename(SampleA = Sample)
df_list <- list(FatherVSOtherM_Data, Age_of_Samples)
FatherVSOtherM_Data <- df_list%>%
  reduce(full_join, by="SampleA")%>%
  filter(SampleB != "NA")%>%
  mutate(SampleB = as.character(as.factor(SampleB)),
    Nestling = ifelse(Ageclass == 'A', SampleB, SampleA))%>%
  select(-c(Ageclass, Nest_Similarity, Age_combination, Sex_combination))
```

4.4.4.1 Analyse

- Fittig model and getting the P-value

```
Model_Male_Nestling <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling),
                               family = beta_family(link = "logit"),
                               data=FatherVSOtherM_Data)
summary(Model_Male_Nestling) #AIC = -6970.3
```

```
##  Family: beta  ( logit )
## Formula:
## BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling)
## Data: FatherVSOtherM_Data
##
##      AIC      BIC   logLik deviance df.resid
##  -3674.4  -3649.4   1842.2  -3684.4     1099
##
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  NestID          (Intercept) 0.08889  0.2981
##  NestID:Nestling (Intercept) 0.13783  0.3712
## Number of obs: 1104, groups:  NestID, 17; NestID:Nestling, 138
##
## Dispersion parameter for beta family (): 73.6
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.744503   0.097097  17.967   <2e-16 ***
## RelationshipOtherM -0.005994   0.316878  -0.019    0.985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-value = 0.985, non-significant. Nestling's preen oil composition is not more similar to that of their father than to another male in the population.

```
#Conversion in linear unit
mu_intercept <- 1.74450
mu_OtherM <- -0.00599
plogis(mu_intercept + mu_OtherM) - plogis(mu_intercept)
```

```
## [1] -0.0007600387
```

The distance Father-Nestling is 0.001 higher than the distance Other male - Nestling in Bray-Curtis units.

- Confidence intervals

```
confint(Model_Male_Nestling)
```

```
##                                        2.5 %    97.5 %    Estimate
## (Intercept)                        1.5541969 1.9348099  1.74450343
## RelationshipOtherM                -0.6270637 0.6150755 -0.00599411
## Std.Dev.(Intercept)|NestID         0.1608617 0.5525809  0.29814280
## Std.Dev.(Intercept)|NestID:Nestling 0.3186839 0.4324849  0.37124923
```

β estimate of the distance to adult males effect: -0.0016 (Beta family unit) Confidence interval: [-0.627 ; 0.615] –> includes "0".

- AIC without Random effects

```
Model_MN_noRandomeff <- glmmTMB(BC_Similarity ~ Relationship,
                                family = beta_family(link = "logit"),
                                data=FatherVSOtherM_Data)
Model_MN_noNestlingID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID),
                                 family = beta_family(link = "logit"),
                                 data=FatherVSOtherM_Data)
Model_MN_noNestID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID:Nestling),
                             family = beta_family(link = "logit"),
                             data=FatherVSOtherM_Data)
anova(Model_Male_Nestling, Model_MN_noRandomeff, Model_MN_noNestlingID, Model_MN_noNestID)
```

```
## Data: FatherVSOtherM_Data
## Models:
## Model_MN_noRandomeff: BC_Similarity ~ Relationship, zi=~0, disp=~1
## Model_MN_noNestlingID: BC_Similarity ~ Relationship + (1 | NestID), zi=~0, disp=~1
## Model_MN_noNestID: BC_Similarity ~ Relationship + (1 | NestID:Nestling), zi=~0, disp=~1
## Model_Male_Nestling: BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling), zi=~0, disp
##                        Df     AIC     BIC logLik deviance    Chisq Chi Df
## Model_MN_noRandomeff    3 -2940.0 -2925.0 1473.0  -2946.0
## Model_MN_noNestlingID   4 -2939.9 -2919.9 1474.0  -2947.9   1.9360      1
## Model_MN_noNestID       4 -3670.7 -3650.7 1839.4  -3678.7 730.7680      0
## Model_Male_Nestling     5 -3674.4 -3649.4 1842.2  -3684.4   5.7023      1
##                        Pr(>Chisq)
## Model_MN_noRandomeff
## Model_MN_noNestlingID     0.16411
## Model_MN_noNestID        < 2e-16 ***
## Model_Male_Nestling      0.01694 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The stronger AIC is for the full model. There is a significant difference between the model without random effects and the model without the NestID effect, but no with the model without NestlingID effect. The NestID effect seems to be the random effect with the most impact.

4.4.5 Adult females VS adult males (other than mother and father) analysis

```
OtherFVSOtherM_Data <- Pairwise_Data%>%
  mutate(Relationship = case_when(
    Sex_combination == 'NF' & Nest_Similarity == '0' ~ 'OtherF',
    Sex_combination == 'NM' & Nest_Similarity == '0' ~ 'OtherM'
  ))%>%
  filter(Relationship != "NA")
Age_of_Samples <- Metadata_Families%>%
  select(Sample, Ageclass)%>%
  rename(SampleA = Sample)
df_list <- list(OtherFVSOtherM_Data, Age_of_Samples)
OtherFVSOtherM_Data <- df_list%>%
  reduce(full_join, by="SampleA")%>%
  filter(SampleB != "NA")%>%
  mutate(SampleB = as.character(as.factor(SampleB)),
    Nestling = ifelse(Ageclass == 'A', SampleB, SampleA))%>%
  select(-c(Ageclass, Nest_Similarity, Age_combination, Sex_combination))
```

4.4.5.1 Analyse

- Fittig model and getting the P-value

```
Model_Adult_Nestling <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling),
                                family = beta_family(link = "logit"),
                                data=OtherFVSOtherM_Data)
summary(Model_Adult_Nestling) #AIC = -6970.3
```

```
##  Family: beta  ( logit )
## Formula:
## BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling)
## Data: OtherFVSOtherM_Data
##
##      AIC      BIC   logLik deviance df.resid
##  -6590.3  -6562.2   3300.1  -6600.3     1999
##
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance  Std.Dev.
##  NestID          (Intercept) 1.353e-09 3.678e-05
##  NestID:Nestling (Intercept) 1.278e-01 3.575e-01
## Number of obs: 2004, groups:  NestID, 1; NestID:Nestling, 69
##
## Dispersion parameter for beta family (): 64.8
##
## Conditional model:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.58315    0.04432   35.72   <2e-16 ***
## RelationshipOtherM  0.14358    0.01477    9.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-value < 2e-16, significant. Nestlings' preen oil composition is not more similar to that of adult males than that of adult females in the population.

```
#Conversion in linear unit
mu_intercept <- 1.58315
mu_OtherM <- 0.14358
plogis(mu_intercept + mu_OtherM) - plogis(mu_intercept)
```

```
## [1] 0.0193435
```

The distance OtherM-Nestling is 0.02 higher than the distance OtherF-Nestling in Bray-Curtis units.

- Confidence intervals

```
confint(Model_Adult_Nestling)
```

```
##                                         2.5 %    97.5 %     Estimate
## (Intercept)                          1.4962823 1.6700188 1.583151e+00
## RelationshipOtherM                   0.1146402 0.1725199 1.435800e-01
## Std.Dev.(Intercept)|NestID           0.0000000       Inf 3.678355e-05
## Std.Dev.(Intercept)|NestID:Nestling 0.3011425 0.4243295 3.574684e-01
```

β estimate of the distance to adult males effect: 0.14 (Beta family unit) Confidence interval: [0.115 ; 0.173] –> does not include "0".

- AIC without Random effects

```
Model_RN_noRandomeff <- glmmTMB(BC_Similarity ~ Relationship,
                                family = beta_family(link = "logit"),
                                data=OtherFVSOtherM_Data)
Model_RN_noNestlingID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID),
                                 family = beta_family(link = "logit"),
                                 data=OtherFVSOtherM_Data)
Model_RN_noNestID <- glmmTMB(BC_Similarity ~ Relationship + (1 | NestID:Nestling),
                             family = beta_family(link = "logit"),
                             data=OtherFVSOtherM_Data)
anova(Model_Adult_Nestling, Model_RN_noRandomeff, Model_RN_noNestlingID, Model_RN_noNestID)
```

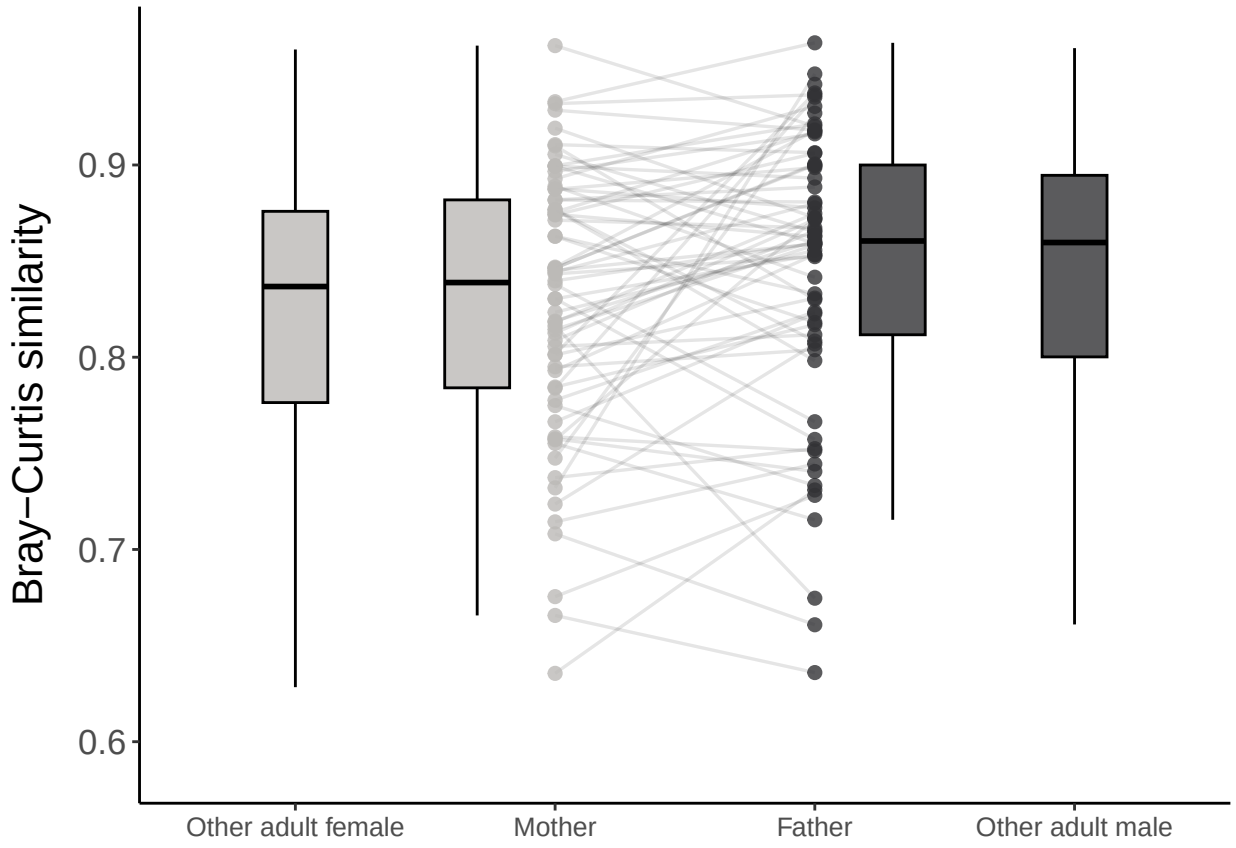
```
## Data: OtherFVSOtherM_Data
## Models:
## Model_RN_noRandomeff: BC_Similarity ~ Relationship, zi=~0, disp=~1
## Model_RN_noNestlingID: BC_Similarity ~ Relationship + (1 | NestID), zi=~0, disp=~1
## Model_RN_noNestID: BC_Similarity ~ Relationship + (1 | NestID:Nestling), zi=~0, disp=~1
## Model_Adult_Nestling: BC_Similarity ~ Relationship + (1 | NestID) + (1 | NestID:Nestling), zi=~0, dis
##                       Df     AIC     BIC logLik deviance  Chisq Chi Df
## Model_RN_noRandomeff   3 -5252.3 -5235.5 2629.2  -5258.3
## Model_RN_noNestlingID  4 -5250.3 -5227.9 2629.2  -5258.3    0.0      1
## Model_RN_noNestID      4 -6592.3 -6569.8 3300.1  -6600.3 1341.9      0
## Model_Adult_Nestling   5 -6590.3 -6562.2 3300.1  -6600.3    0.0      1
##                       Pr(>Chisq)
## Model_RN_noRandomeff
## Model_RN_noNestlingID          1
## Model_RN_noNestID         <2e-16 ***
## Model_Adult_Nestling           1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The stronger AIC is for the model without NestID. This is the only model that is significantly different from the one with no random effect. NestID seem to be the only important random effect here. But note that the full model has almost the same AIC.

4.4.6 Plot

```
# Prepare data for plotting
Data_plot <- Pairwise_Data%>%
  mutate(Relationship = case_when(
    Sex_combination == 'NM' & Nest_Similarity == '1' ~ 'Father',
```

```
    Sex_combination == 'NF' & Nest_Similarity == '1' ~ 'Mother',
    Sex_combination == 'NF' & Nest_Similarity == '0' ~ 'Other adult female',
    Sex_combination == 'NM' & Nest_Similarity == '0' ~ 'Other adult male',
  ))%>%
  filter(Relationship != "NA")
Age_of_Samples <- Metadata_Families%>%
  select(Sample, Ageclass)%>%
  rename(SampleA = Sample)
df_list <- list(Data_plot, Age_of_Samples)
Data_plot <- df_list%>%
  reduce(full_join, by="SampleA")%>%
  filter(SampleB != "NA")%>%
  mutate(SampleB = as.character(as.factor(SampleB)),
         Nestling = ifelse(Ageclass == 'A', SampleB, SampleA))%>%
  select(-c(Ageclass, Nest_Similarity, Age_combination, Sex_combination))

# Plot
ggplot(Data_plot, aes(x = Relationship, y = BC_Similarity)) +
  geom_boxplot(data = Data_plot %>% filter(Relationship == "Other adult female"), outlier.shape = NA, wi
  geom_boxplot(data = Data_plot %>% filter(Relationship == "Mother"), position = position_nudge(x = -0.3
  geom_boxplot(data = Data_plot %>% filter(Relationship == "Father"), position = position_nudge(x = 0.3)
  geom_boxplot(data = Data_plot %>% filter(Relationship == "Other adult male"), outlier.shape = NA, widt
  geom_line(data = Data_plot %>% filter(Relationship == "Father" | Relationship == "Mother"), aes(group
  geom_point(data = Data_plot %>% filter(Relationship == "Mother" | Relationship == "Father"), aes(colo
  scale_color_manual(values = c("#323235","#BCBAB7")) +
  ylab("Bray-Curtis similarity") +
  theme(legend.position = "none",
        axis.title.y=element_text(size=16, margin = margin(t = 0, r = 10, b = 0, l = 0)),
        axis.text.y = element_text(size=13),
        axis.text.x = element_text(size=10),
        axis.title.x = element_blank()) +
  scale_x_discrete(limits = c("Other adult female", "Mother", "Father", "Other adult male"))
```



5 Control analysis including the four clear outliers

The sample sizes for the analysis of the effect of breeding stage and individual ID in females is slightly lower than in the pre-registration (repeated samples for 29 females instead of 33). Indeed, we discarded four samples from incubating females for which the alignment did not accurately reflect the chromatogram. But to make sure that these samples would not affect our conclusions, we re-made the breeding-stage and individual identity analysis in females including these outliers.

Data = 33 individual females (66 samples) sampled both during the incubation and nestling-rearing period

```
# Subset of the Metadata for the breeding stage and individual identity analyses:
F_sampled_twice_out <- Metadata%>%   # 58 samples
  filter(f_sampled_twice==1)%>%
  mutate(Breeding_Stage = as.factor(as.character(Breeding_Stage)))%>%
  select(-c(pair_brood, Partner_Connected_to_Outlier, Outliers, F_Connected_to_Outlier, Families, Famili
```

5.0.1 NMDS plot

Step 1: Building a Bray-Curtis matrix

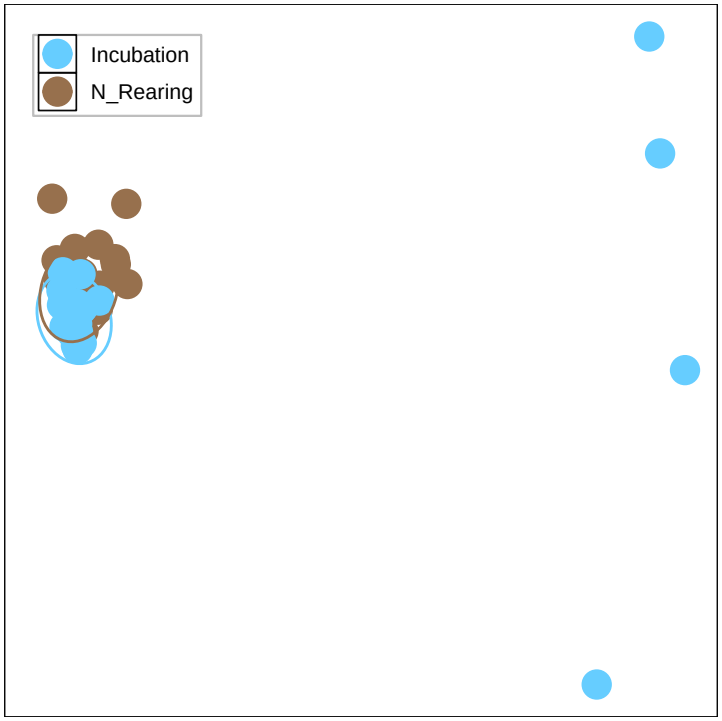
```
ChemdataBS <- Chemdata%>%
  filter(Sample %in% F_sampled_twice_out$Sample)%>%
  select(-Sample)
bc <- metaMDS(ChemdataBS, distance = "bray")
```

Step 2: Checking the stress

```
bc$stress
```

```
## [1] 0.02927406
```

Step 3: Plot the NMDS, here by breeding stage

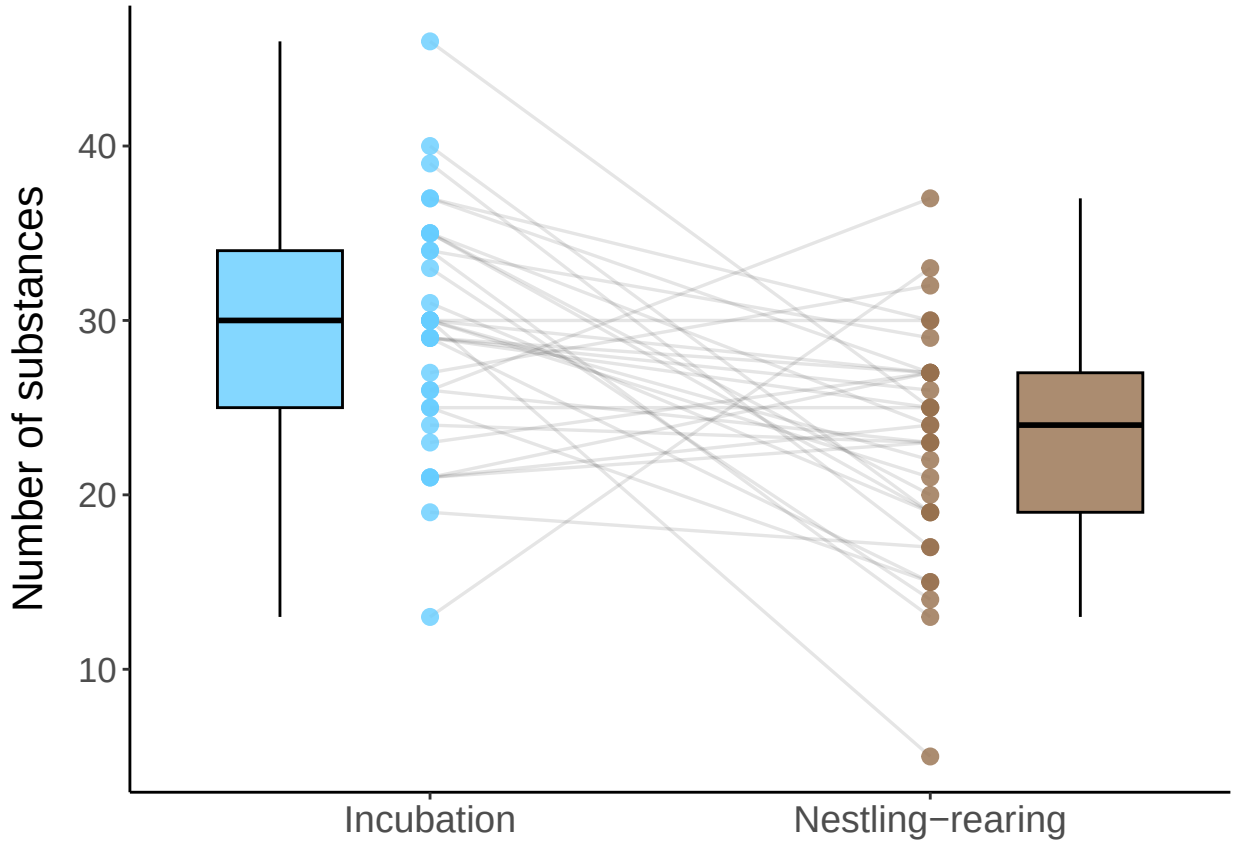


5.0.2 Richness

a. Visualisation of the breeding stage difference (within individuals) in terms of chemical richness.

```
theme_set(theme_classic())
ggplot(F_sampled_twice_out, aes(x=Breeding_Stage, y=Richness))+
  geom_boxplot(data = F_sampled_twice_out %>% filter(Breeding_Stage=="Incubation"),
               aes(x=Breeding_Stage, y=Richness),
               position=position_nudge(x=-0.3), outlier.shape = NA, width = 0.25,lwd = 0.5,
               alpha = 0.8, colour="black", fill = "#66cdff")+
  geom_boxplot(data = F_sampled_twice_out %>% filter(Breeding_Stage=="N_Rearing"), lwd = 0.5,
               aes(x=Breeding_Stage, y=Richness),
               position=position_nudge(x=0.3), outlier.shape = NA, width = 0.25,
               alpha = 0.8, colour="black", fill = "#97704d")+
  geom_line(aes(group=Individual_ID),alpha = 0.1, size = 0.6, position=position_dodge(0), show.legend =
  geom_point(aes(color = Breeding_Stage), alpha = 0.8, size=2.5,show.legend = FALSE)+
```

```
  scale_color_manual(values = c("#66cdff", "#97704d")) +
  scale_x_discrete(labels=c("Incubation","Nestling-rearing")) +
  ylab("Number of substances") +
  theme(axis.title.y=element_text(size=16, margin = margin(t = 0, r = 10, b = 0, l = 0)),
        axis.text.y = element_text(size=13),
        axis.text.x = element_text(size=14),
        axis.title.x = element_blank())
```



b. Analysis

- Fitting the model

```
LMM_B_Stage_R <- lme4::lmer(formula = "Richness ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampl
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(LMM_B_Stage_R)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Richness ~ Breeding_Stage + (1 | Individual_ID)
##    Data: F_sampled_twice_out
##
## REML criterion at convergence: 431.7
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6985 -0.6032  0.0771  0.5941  2.4717
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept)  0.00    0.000
##  Residual                  44.64    6.682
## Number of obs: 66, groups:  Individual_ID, 33
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)                29.485      1.163  25.350
## Breeding_StageN_Rearing    -6.455      1.645  -3.924
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Richness ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampled_tw
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Richness ~ Breeding_Stage + (1 | Individual_ID)"
##    Data: F_sampled_twice_out
##
## REML criterion at convergence: 431.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6985 -0.6032  0.0771  0.5941  2.4717
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept)  0.00    0.000
##  Residual                  44.64    6.682
## Number of obs: 66, groups:  Individual_ID, 33
##
## Fixed effects:
##                          Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)                29.485      1.163 64.000  25.350  < 2e-16 ***
## Breeding_StageN_Rearing    -6.455      1.645 64.000  -3.924 0.000215 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_B_Stage_R, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group         term            estim~1 std.e~2 stati~3 conf.~4 conf.~5
##   <chr>    <chr>         <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>          (Intercept)       29.5     1.16   25.3    27.1    31.6
## 2 fixed    <NA>          Breeding_Stage~   -6.45    1.64   -3.92   -9.71   -3.53
## 3 ran_pars Individual_ID sd__(Intercept)    0      NA      NA       0       4.08
## 4 ran_pars Residual      sd__Observation    6.68   NA      NA       4.97    7.65
## # ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
## #   4: conf.low, 5: conf.high
```

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_B_Stage_R, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.1915 0.0547   0.3728   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

- Finding the repeatability (random effect)

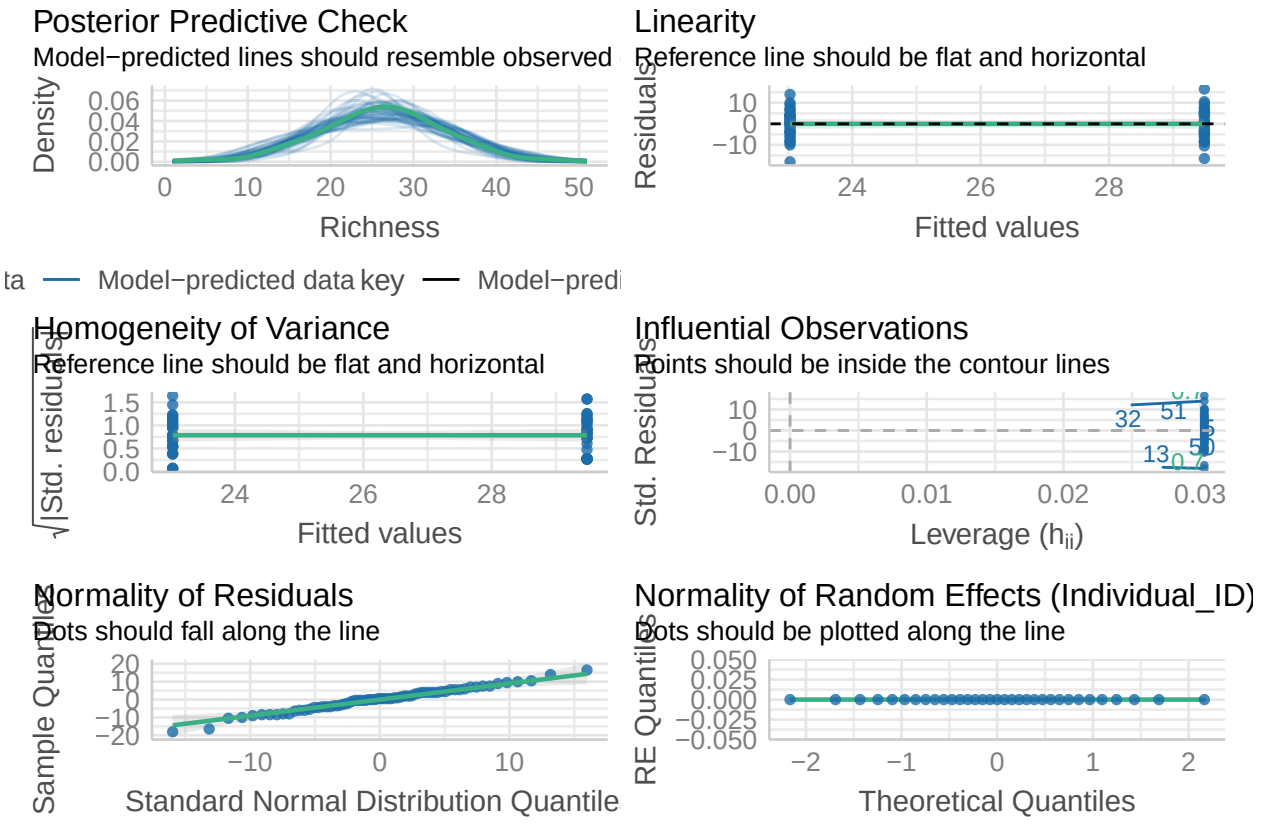
```
rpt(Richness ~ Breeding_Stage + (1 | Individual_ID),
    grname = "Individual_ID",
    data = F_sampled_twice_out, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Individual_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Individual_ID
## R  = 0
```

```
## SE = 0.105
## CI = [0, 0.353]
## P  = 1 [LRT]
##      1 [Permutation]
```

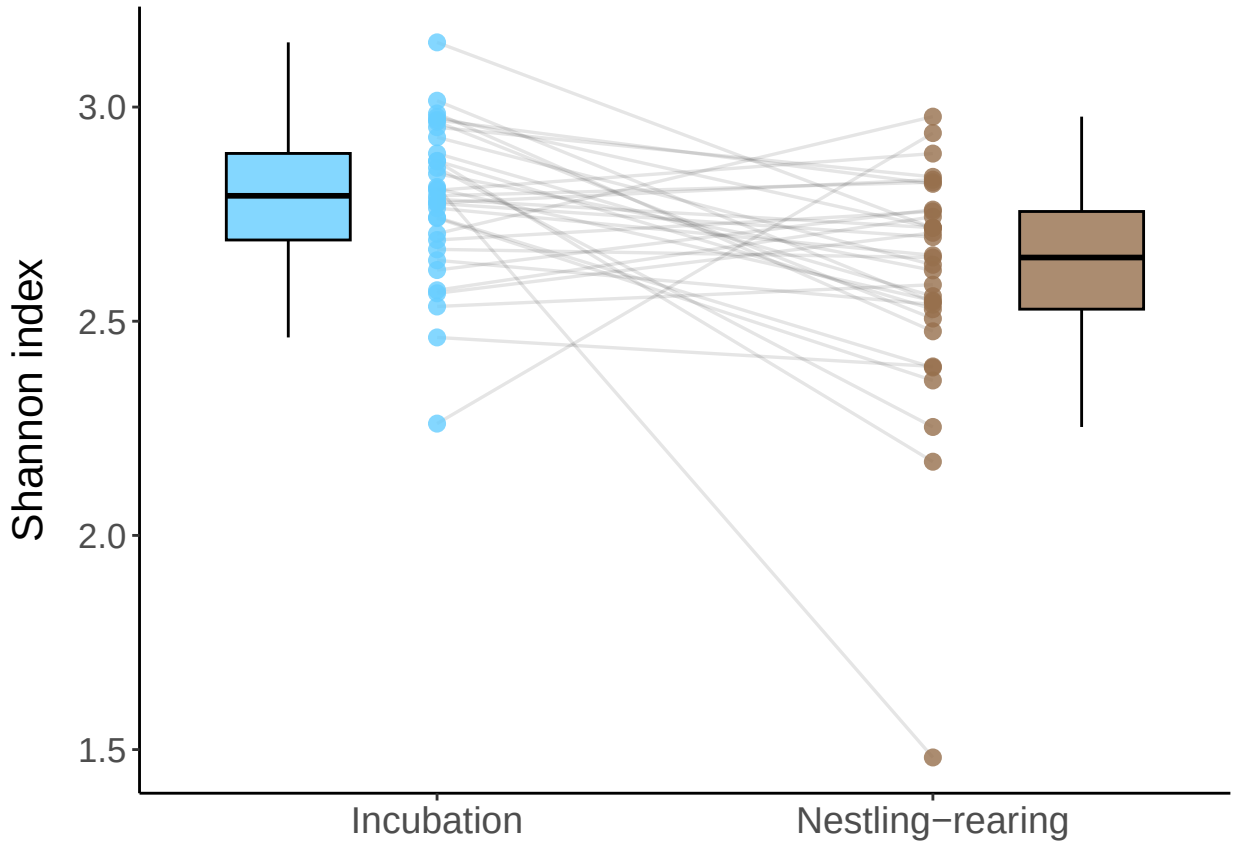
- Model diagnostic

```
check_model(LMM_B_Stage_R)
```



5.0.3 Diversity

a. Visualisation of the breeding stage difference (within individuals) in terms of Shannon diversity.



b. Analysis

- Fitting the model

```
LMM_B_Stage_D <- lme4::lmer(formula = "Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)",  data = F
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(LMM_B_Stage_D)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)
##    Data: F_sampled_twice_out
##
## REML criterion at convergence: 1.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8131 -0.3820  0.1091  0.6288  1.6191
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.00000  0.0000
##  Residual                  0.05409  0.2326
```

```
## Number of obs: 66, groups:  Individual_ID, 33
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)               2.78214    0.04049  68.716
## Breeding_StageN_Rearing -0.18112    0.05726  -3.163
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampl
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Shannon_Index ~ Breeding_Stage + (1 | Individual_ID)"
##    Data: F_sampled_twice_out
##
## REML criterion at convergence: 1.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8131 -0.3820  0.1091  0.6288  1.6191
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.00000  0.0000
##  Residual                  0.05409  0.2326
## Number of obs: 66, groups:  Individual_ID, 33
##
## Fixed effects:
##                          Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)               2.78214    0.04049 64.00000  68.716  < 2e-16 ***
## Breeding_StageN_Rearing -0.18112    0.05726 64.00000  -3.163  0.00239 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_B_Stage_D, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group         term            estim~1 std.e~2 stati~3 conf.~4 conf.~5
##   <chr>    <chr>         <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>          (Intercept)       2.78   0.0405   68.7    2.70   2.87
## 2 fixed    <NA>          Breeding_Stage~  -0.181  0.0573   -3.16  -0.305 -0.0598
## 3 ran_pars Individual_ID sd__(Intercept)   0     NA        NA      0      0.146
## 4 ran_pars Residual      sd__Observation   0.233 NA        NA      0.174  0.266
## # ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
## #   4: conf.low, 5: conf.high
```

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_B_Stage_D, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2     CI_lower CI_upper nboot ndf
##  0.1334 0.0162   0.3106   1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

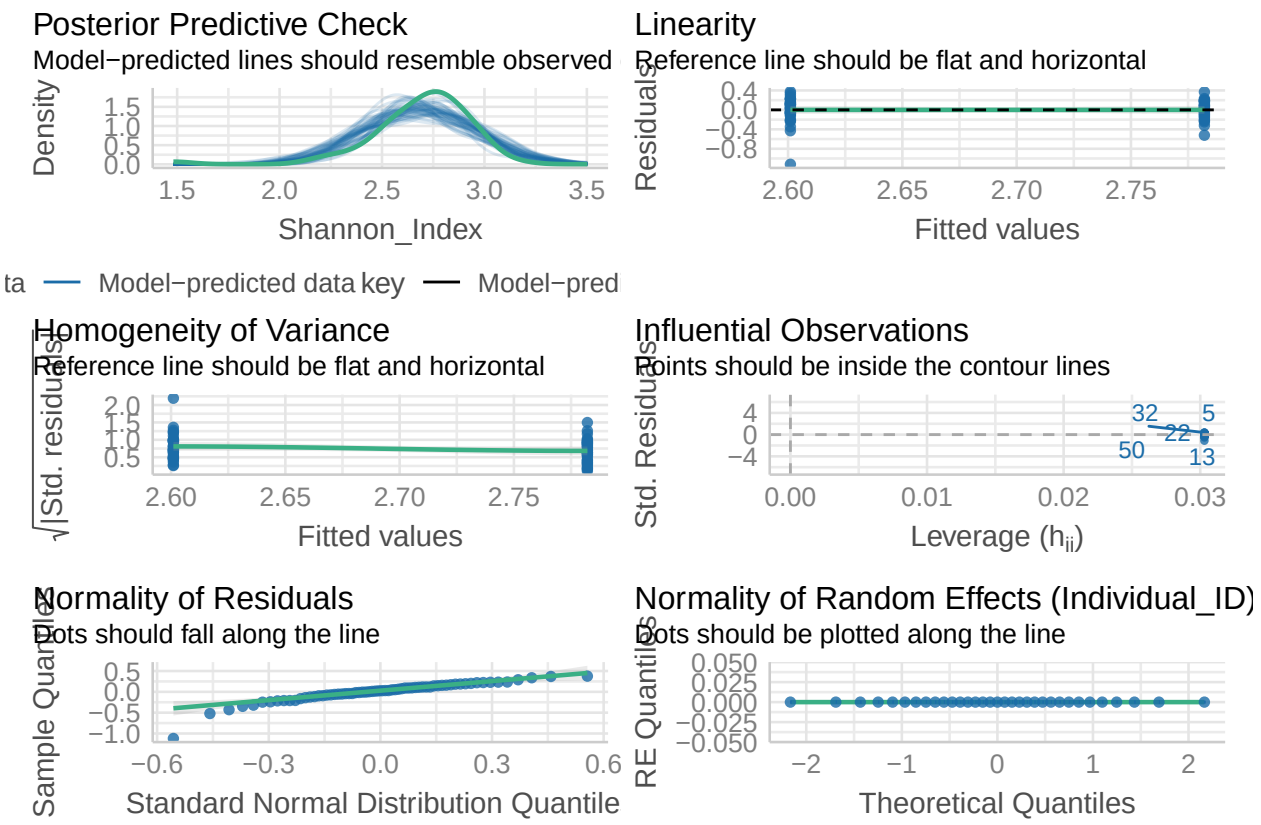
- Finding the repeatability (random effect)

```
rpt(Shannon_Index ~ Breeding_Stage + (1 | Individual_ID),
    grname = "Individual_ID",
    data = F_sampled_twice_out, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Individual_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Individual_ID
## R  = 0
## SE = 0.105
## CI = [0, 0.355]
## P  = 1 [LRT]
##      1 [Permutation]
```

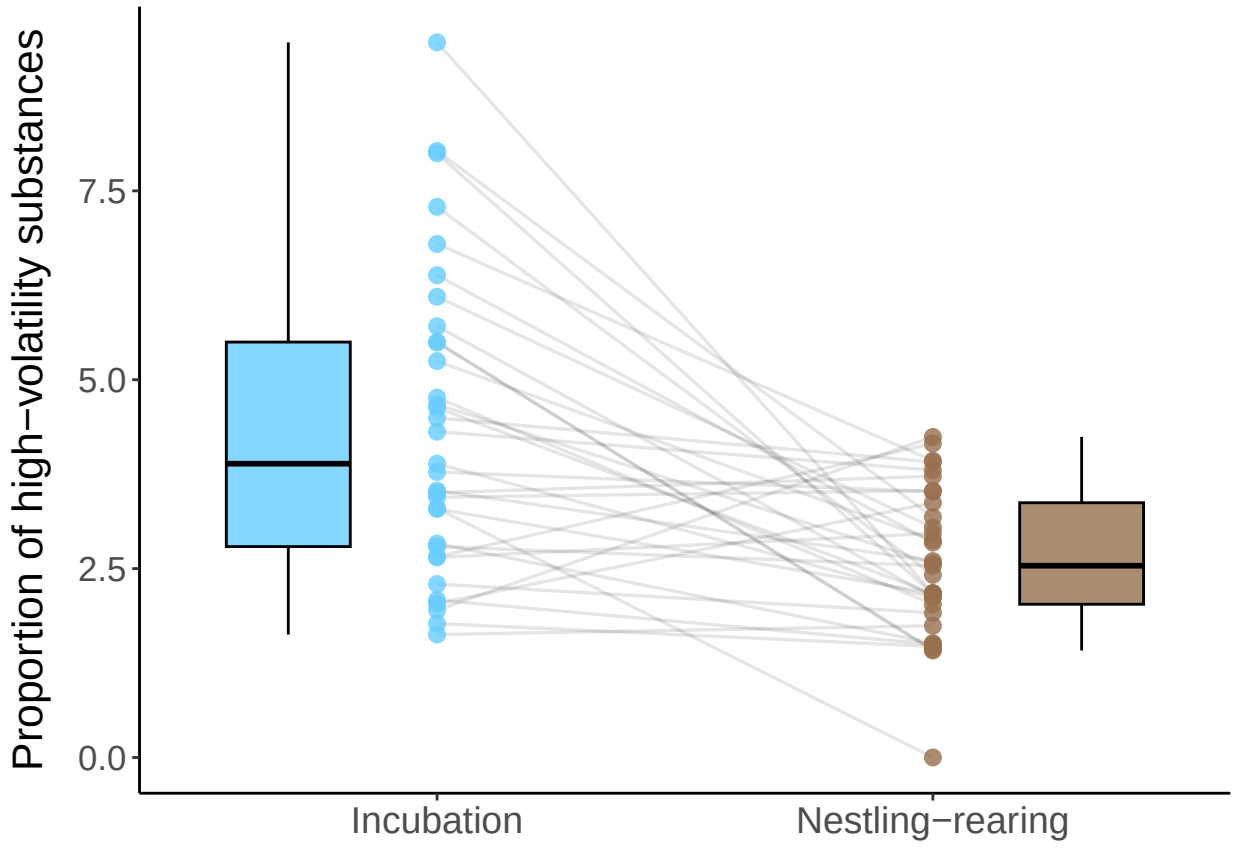
- Model diagnostic

```
check_model(LMM_B_Stage_D)
```



5.0.4 Volatility

a. Visualisation of the breeding stage difference (within individuals) in terms of volatility.



b. Analysis

- Fitting the model

```
LMM_B_Stage_V <- lme4::lmer(formula = "Volatility ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sa
summary(LMM_B_Stage_V)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Volatility ~ Breeding_Stage + (1 | Individual_ID)
##    Data: F_sampled_twice_out
##
## REML criterion at convergence: 247.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6974 -0.6350 -0.1028  0.6038  3.1800
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.0644   0.2538
##  Residual                  2.4485   1.5648
## Number of obs: 66, groups:  Individual_ID, 33
##
## Fixed effects:
##                         Estimate Std. Error t value
```

```
## (Intercept)               4.3729     0.2759  15.847
## Breeding_StageN_Rearing  -1.7922     0.3852  -4.652
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.698
```

- Finding the P-value for the fixed effect

```
summary(lmerTest::lmer(formula = "Volatility ~ Breeding_Stage + (1 | Individual_ID)",  data = F_sampled_
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: "Volatility ~ Breeding_Stage + (1 | Individual_ID)"
##    Data: F_sampled_twice_out
##
## REML criterion at convergence: 247.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6974 -0.6350 -0.1028  0.6038  3.1800
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Individual_ID (Intercept) 0.0644   0.2538
##  Residual                  2.4485   1.5648
## Number of obs: 66, groups:  Individual_ID, 33
##
## Fixed effects:
##                          Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                4.3729     0.2759 63.9580  15.847  < 2e-16 ***
## Breeding_StageN_Rearing   -1.7922     0.3852 32.0000  -4.652 5.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## Brdng_StN_R -0.698
```

- Finding the β estimate and it's confidence interval (fixed effect)

```
tidy(LMM_B_Stage_V, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group         term            estim~1 std.e~2 stati~3 conf.~4 conf.~5
##   <chr>    <chr>         <chr>             <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 fixed    <NA>          (Intercept)       4.37    0.276   15.8     3.85   4.93
## 2 fixed    <NA>          Breeding_Stage~  -1.79    0.385   -4.65   -2.54  -0.982
## 3 ran_pars Individual_ID sd__(Intercept)   0.254  NA       NA       0      0.974
## 4 ran_pars Residual      sd__Observation   1.56   NA       NA       1.19   1.80
## # ... with abbreviated variable names 1: estimate, 2: std.error, 3: statistic,
## #   4: conf.low, 5: conf.high
```

- Finding the marginal R^2 (fixed effect)

```
partR2(LMM_B_Stage_V, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##  R2    CI_lower CI_upper nboot ndf
##  0.245 0.0953   0.426    1000  2
##
## ----------
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

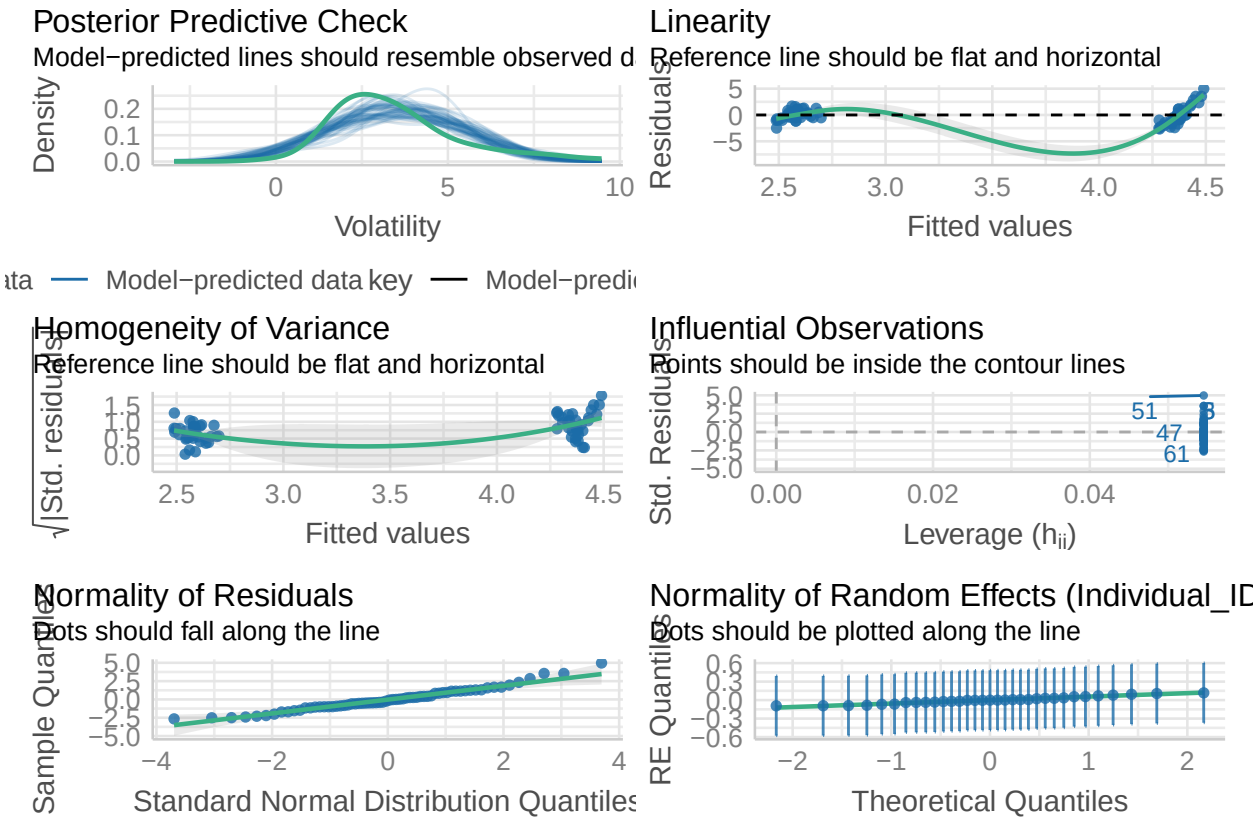
- Finding the repeatability (random effect)

```
rpt(Volatility ~ Breeding_Stage + (1 | Individual_ID),
    grname = "Individual_ID",
    data = F_sampled_twice_out, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for Individual_ID :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for Individual_ID
## R  = 0.026
## SE = 0.113
## CI = [0, 0.372]
## P  = 0.441 [LRT]
##      0.489 [Permutation]
```

- Model diagnostic

```
check_model(LMM_B_Stage_V)
```


Posterior Predictive Check
Model−predicted lines should resemble observed d
Density
Volatility
ata
Model−predicted data
key
Model−predi
Linearity
Reference line should be flat and horizontal
Residuals
Fitted values
Homogeneity of Variance
Reference line should be flat and horizontal
√|Std. residuals|
Fitted values
Influential Observations
Points should be inside the contour lines
Std. Residuals
Leverage (hii)
51
47
61
Normality of Residuals
Dots should fall along the line
Sample Quantiles
Standard Normal Distribution Quantiles
Normality of Random Effects (Individual_ID
Dots should be plotted along the line
RE Quantiles
Theoretical Quantiles