

Preen oil composition of Pied Flycatchers differs between sexes and breeding stages but is similar between partners

R code

M. Gilles, R.W. Fokkema, P. Korsten, B.A. Caspers & T. Schmoll

Contents

1	Data	2
1.1	Metadata	2
1.2	Chemical data	2
1.2.1	Alignment	2
1.2.2	Transformation	7
1.2.3	Abundance of most common substance	7
1.2.4	Richness and diversity	7
1.2.5	Volatility	8
1.3	Data selection	9
1.3.1	Samples without preen oil	9
1.3.2	Samples with low concentration	9
1.3.3	Data for sex and pair analysis	11
1.3.4	Data for season and individual analysis	11
1.3.5	Concentration differences between groups	12
2	Analysis	15
2.1	Sex and pair analysis	15
2.1.1	Richness (S)	15
2.1.2	Diversity (H)	18
2.1.3	Proportion of high-volatility substances	23
2.1.4	Proportion of low-volatility substances	28
2.2	Season and pair analysis	31
2.2.1	Richness (S)	31

2.2.2	Diversity (H)	34
2.2.3	Proportion of high-volatility substances	37
2.2.4	Proportion of low-volatility substances	40

Required packages

```
library(GCalignR) # to align chromatograms
library(vegan) # to calculate chemical richness and diversity
library(dplyr) # for manipulation of data sets
library(ggplot2) # for plotting
library(lme4) # to run linear mixed models
library(lmerTest) # to run linear mixed models and get p-values from mixed models
library(rptR) # to get repeatabilities from mixed models
library(broom.mixed) # to get confidence intervals from mixed models
library(partR2) # to get R2 and confidence intervals of fixed effects from mixed models
library(performance) # for model diagnostics
library(qqplotr) # so that package performance can work completely
library(lattice) # for data visualization
library(pracma) # to get x for a specific y from the cumulative abundance curve
library(tidyverse)
```

1 Data

1.1 Metadata

Import and prepare metadata

```
# import metadata
metadata<-read.csv("raw_metadata.csv")

# set sampleIDs as row names
metadata <- data.frame(metadata, row.names = 1)

# make factors of variables not yet designated as factors
str(metadata)
metadata$ringnr<-as.factor(metadata$ringnr)
metadata$sampletime<-as.factor(metadata$sampletime)
str(metadata)

# change order of levels in factors
metadata$breeding_stage <- factor(metadata$breeding_stage, levels = c("Before", "After"))
```

1.2 Chemical data

1.2.1 Alignment

Find optimal reference sample

```
choose_optimal_reference("raw_chemdata.txt", rt_col_name = "RT")
```

```
## $sample  
## [1] "LI150"  
##  
## $score  
## [1] 0.317844
```

Prepare a vector with the name of the GC control samples (“DCM_”) and environmental control samples (“LI”) (substances detected in GC control samples and environmental control samples will be removed during the alignment)

```
blanks.and.controls <- c("DCM_026", "DCM_101", "DCM_76", "LI106", "LI71", "LI90", "LI133")
```

Alignment

```
aligned <- align_chromatograms(  
  data = "raw_chemdata.txt",  
  # raw chromatographic data  
  rt_col_name = "RT",  
  reference = "LI150",  
  # obtained from the function 'choose_optimal_reference()'  
  blanks = blanks.and.controls,  
  # 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,  
  # expected linear drift  
  max_diff_peak2mean = 0.025,  
  # allowed RT difference of a peak with the mean of the corresponding row  
  min_diff_peak2peak = 0.05,  
  # expected minimum RT difference among homologous substances  
  permute = F,  
  # keep the order of samples constant between different alignments  
  write_output = c("area"))
```

Summary of the alignment procedure

```
print(aligned)
```

```
## Summary of Peak Alignment running align_chromatograms  
## Input: raw_chemdata.txt  
## Start: 2022-03-24 15:46:29 Finished: 2022-03-24 15:54:06  
##  
## Call:  
## GCalignR::align_chromatograms(data=raw_chemdata.txt, rt_col_name=RT,  
## write_output=(area), reference=LI150, max_linear_shift=0,  
## max_diff_peak2mean=0.025, min_diff_peak2peak=0.05, delete_single_peak=TRUE,  
## remove_empty=TRUE, permute=F, sep=\t, rt_cutoff_low=NULL,
```

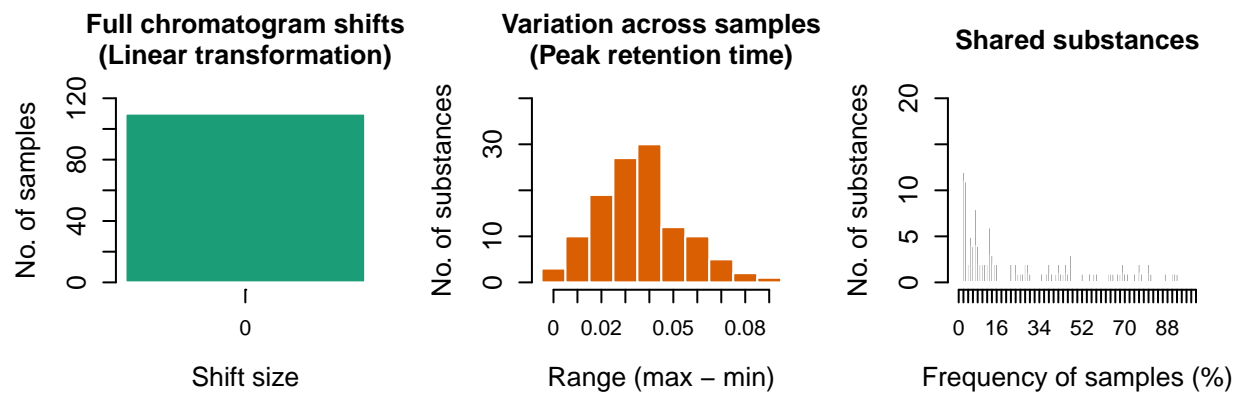
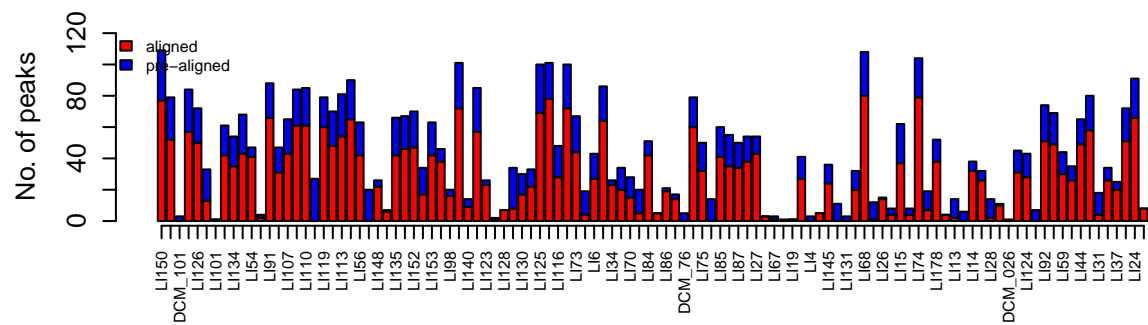
```

##   rt_cutoff_high=NULL, ...=)
##
## Summary of scored substances:
##   total   blanks singular retained
##   182     39      24      119
##
## In total 182 substances were identified among all samples. 39 substances were
## present in blanks. The corresponding peaks as well as the blanks were removed
## from the data. 24 substances were present in just one single sample and were
## removed. 119 substances are retained after all filtering steps.
##
## Sample overview:
## The following 110 samples were aligned to the reference 'LI150':
## LI150, LI97, LI81, LI126, LI138, LI101, LI144, LI134, LI112, LI54, LI132, LI91,
## LI105, LI107, LI104, LI110, LI119, LI46, LI113, LI48, LI56, LI148, LI94, LI135,
## LI146, LI152, LI147, LI153, LI122, LI98, LI151, LI140, LI142, LI123, LI69,
## LI128, LI89, LI130, LI139, LI125, LI83, LI116, LI82, LI73, LI18, LI6, LI40,
## LI34, LI72, LI70, LI8, LI84, LI102, LI86, LI114, LI36, LI75, LI85, LI115, LI87,
## LI66, LI27, LI79, LI67, LI19, LI12, LI141, LI145, LI95, LI68, LI10, LI26, LI1,
## LI15, LI111, LI74, LI22, LI178, LI7, LI13, LI14, LI42, LI28, LI93, LI96, LI124,
## LI92, LI127, LI59, LI129, LI44, LI35, LI31, LI29, LI37, LI38, LI24, LI39
##
## For further details type:
##   'gc_heatmap(aligned)' to retrieve heatmaps
##   'plot(aligned)' to retrieve further diagnostic plots

```

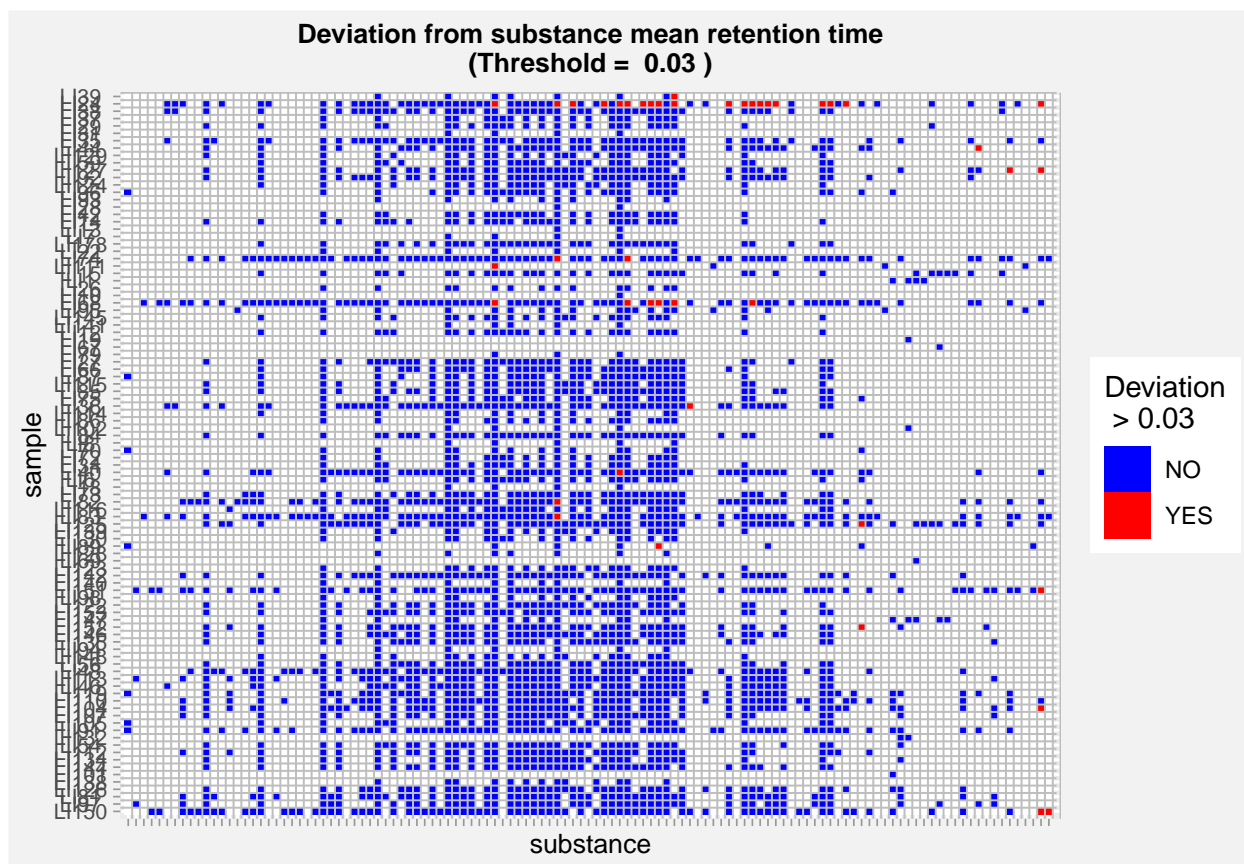
Diagnostic plots

```
plot(aligned,which_plot = "all")
```



Heatmap of aligned chromatographic data

```
gc_heatmap(aligned, threshold = 0.03)
```



Save the aligned chromatographic data

```
save(aligned, file = "aligned_chromdata.RData")
```

Load the aligned chromatographic data

```
# automatic method
aligned
```

```
# manual method (load the data from the txt file created by GCalignR)
chemdata.manual <- read.table("raw_chemdata_area.txt",header = F)
samplenames <- chemdata.manual[1,]
rownames(chemdata.manual) <- as.matrix(chemdata.manual[,1])
chemdata.manual <- chemdata.manual[,-1]
colnames(chemdata.manual) <- as.matrix(chemdata.manual[1,])
chemdata.manual <- chemdata.manual[-1,]
chemdata.manual <- as.data.frame(t(chemdata.manual))
str(chemdata.manual)
chemdata.num <- data.frame(lapply(chemdata.manual,
                                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.manual <- chemdata.num
```

In total, 98 samples were aligned and 119 substances were detected.

1.2.2 Transformation

Standardise to get the relative abundances of substances within each sample

```
chemdata.norm <- norm_peaks(aligned, conc_col_name = "area", rt_col_name = "RT",
                           out = "data.frame") # proportions range between 0 und 100 (%)
```

Log-transform ($\log(x+1)$) relative abundances to increase the weight of the numerous low-abundance substances and decrease the weight of the few high-abundance substances

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

Make sure that dataframes 'chemdata' and 'metadata' have the same rows

```
metadata <- metadata[rownames(metadata) %in% rownames(chemdata),]
chemdata <- chemdata[rownames(chemdata) %in% rownames(metadata),]
chemdata <- chemdata[match(rownames(metadata), rownames(chemdata)),]
```

1.2.3 Abundance of most common substance

Find the most common substance

```
max(colSums(chemdata.manual != 0))
colSums(chemdata.manual != 0)
```

The substance with a retention time of 17.41 min is the most common substance (present in 91 samples).

Get the abundance of the most common substance for each sample

```
abundance_mostcommonpeak <- chemdata.manual[, "X17.4091868131868", drop = F]
colnames(abundance_mostcommonpeak) <- c("abundance_mostcommonpeak")
```

Add it to the metadata

```
metadata <- merge(metadata, abundance_mostcommonpeak, by = 0)
rownames(metadata) <- as.matrix(metadata[,1])
metadata <- metadata[,-1]
```

We will use the abundance of the most common substance as a proxy of the concentration of the samples.

1.2.4 Richness and diversity

Calculate richness (S)

```
S <- specnumber(chemdata)
```

Calculate Shannon diversity (H)

```
H <- diversity(chemdata)
```

Add richness and diversity to the metadata

```
sampleID <- rownames(chemdata)
diversity <- data.frame(S = S, H = H)
diversity <- diversity[match(rownames(metadata), rownames(diversity)),]
metadata <- cbind(metadata, diversity)
```

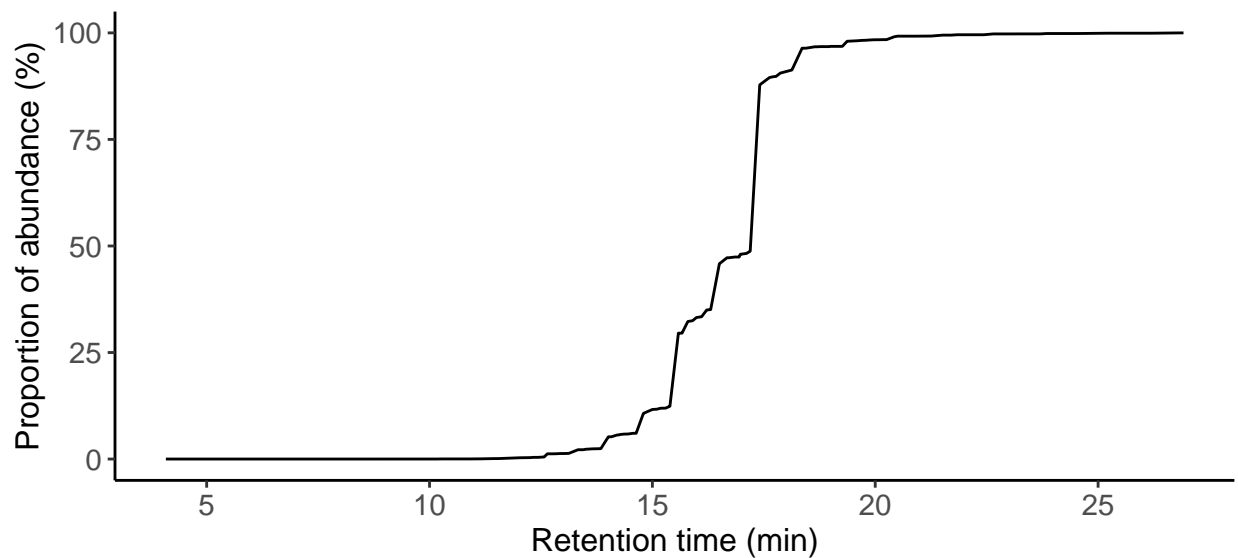
1.2.5 Volatility

Calculate the cumulative abundance curve for each sample

```
prop.cum.abundances <- t(apply(chemdata.norm, 1, cumsum))
```

Example of the cumulative abundance curve of one sample

```
RT <- as.numeric(colnames(prop.cum.abundances))
prop <- prop.cum.abundances["LI150",] # sample LI150
data.plot <- data.frame(RT, prop)
theme_set(theme_classic())
ggplot(data = data.plot, aes(x = RT, y = prop)) +
  geom_line(data = data.plot, aes(x = RT, y = prop)) +
  scale_x_continuous(n.breaks = 8) +
  labs(x = "Retention time (min)", y = "Proportion of abundance (%)") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 12))
```



Get the proportion of the chromatograms before RT = 15.58 min (“high-volatility substances”) and after RT = 17.41 min (“low-volatility substances”)


```

volatility <- prop.cum.abundances[,c("15.3931194029851", "17.4091868131868"), drop = F]
# we use peak 15.39 to have the cumulative abundance BEFORE peak 15.58
# we use peak 16.50 to have the cumulative abundance AFTER peak 17.41
colnames(volatility) <- c("prop.cum.ab.15.39", "prop.cum.ab.17.41")
volatility <- as.data.frame(volatility)
volatility$prop.highvolatility <- volatility$prop.cum.ab.15.39
volatility$prop.lowvolatility <- 100-volatility$prop.cum.ab.17.41
volatility <- volatility[, -c(1,2)]

```

Add it to the metadata

```

metadata <- merge(metadata, volatility, by = 0)
rownames(metadata) <- as.matrix(metadata[,1])
metadata <- metadata[, -1]

```

1.3 Data selection

1.3.1 Samples without preen oil

Remove samples which do not seem to contain preen oil substances (by visual inspection of the chromatograms)

```

metadata <- metadata[metadata$quality_chrom == "good",]
metadata$quality_chrom <- droplevels(metadata$quality_chrom)
nrow(metadata)

```

```
## [1] 77
```

21 samples do not contain preen oil and are discarded. 77 samples are retained for the analysis.

1.3.2 Samples with low concentration

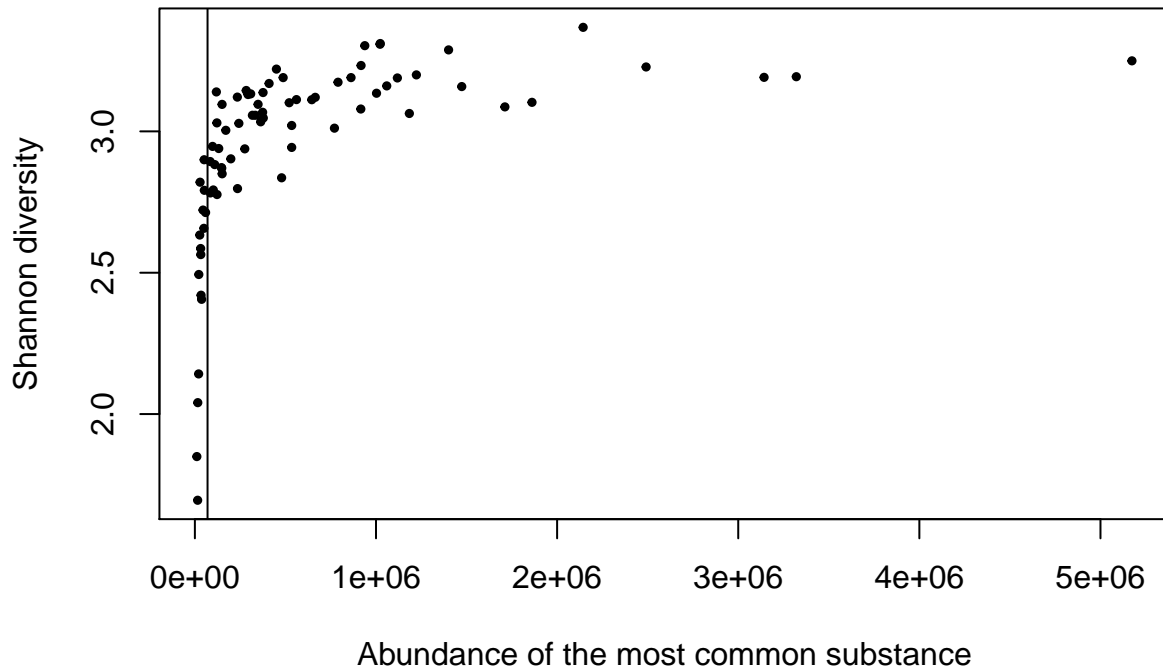
Some samples seem to have a limited richness/diversity. These samples appear too be not concentrated enough for low abundance substances to be detected, resulting in a lower richness/diversity (concentration bias).

Plot Shannon diversity against the concentration (abundance of the most common substance) of the samples

```

plot(metadata$H~metadata$abundance_mostcommonpeak, pch = 19, cex = 0.5,
      col=metadata$quality_chrom, ylab = "Shannon diversity",
      xlab = "Abundance of the most common substance")
abline(v=70000, col="black")

```



The curved pattern shows that diversity drops below a threshold of concentration (abundance of the most common peak = 70000). We consider that samples below that value are “low-concentration” samples and reflect poorly the chemical composition of the preen oil they contain (non-detection of low-abundance substances, underestimated diversity/richness).

```
metadata.lowconc <- metadata[metadata$abundance_mostcommonpea < 70000,]
nrow(metadata.lowconc)
```

```
## [1] 16
```

In total, 16 samples appear to have a low concentration of preen oil (abundance of the most common peak < 70000).

PERMANOVA analyses will be conducted on two data sets:

- data set 1 (complete) contains all the samples (77 samples)
- data set 2 (reduced) contains only the samples that have a good concentration (i.e. most common substance > 70000) (61 samples)

GLMM analyses will be conducted on the reduced data set only, because we used diversity to find a selection threshold, and should therefore not compare the results (e.g. differences in diversity) obtained with the complete data set to those obtained with reduced data set.

1.3.3 Data for sex and pair analysis

We include only complete pairs (i.e. where both the female and the male were sampled) in order to have a perfectly balanced design.

Subset data to keep only samples during nestling rearing from complete pairs (female and male)

```
data.sex <- metadata
data.sex <- data.sex[data.sex$breeding_stage == "After",]
data.sex <- data.sex[ ave(1:nrow(data.sex), data.sex$nestbox_factor,
                        FUN=length) > 1 , ]
data.sex$nestbox_factor <- droplevels(data.sex$nestbox_factor)
```

Data set 1 (complete)

```
data.sex.1 <- data.sex
table(data.sex.1$sex)
```

```
##
## Female    Male
##      17      17
```

$N = 34$ samples from 17 pairs (17 females and 17 males)

Data set 2 (reduced)

```
data.sex.2 <- data.sex[data.sex$abundance_mostcommonpeak > 70000,]
data.sex.2 <- data.sex.2[ ave(1:nrow(data.sex.2), data.sex.2$nestbox_factor,
                            FUN=length) > 1 , ]
data.sex.2$nestbox_factor <- droplevels(data.sex.2$nestbox_factor)
table(data.sex.2$sex)
```

```
##
## Female    Male
##      11      11
```

$N = 22$ samples from 11 pairs (11 females and 11 males)

1.3.4 Data for season and individual analysis

We include only females that were sampled twice (i.e. during incubation and during nestling rearing) in order to have a perfectly balanced design.

Subset data to keep only samples from females that were sampled twice (during incubation and during nestling rearing)

```
data.season <- metadata
data.season <- data.season[data.season$sex == "Female",]
data.season <- data.season[ ave(1:nrow(data.season), data.season$ringnr,
                              FUN=length) > 1 , ]
data.season$ringnr <- droplevels(data.season$ringnr)
```

Data set 1 (complete)

```
data.season.1 <- data.season
table(data.season.1$breeding_stage)
```

```
##
## Before After
##      12      12
```

N = 24 samples from 12 females sampled twice (12 during incubation and 12 during nestling rearing)

Data set 2 (reduced)

```
data.season.2 <- data.season[data.season$abundance_mostcommonpeak > 70000,]
data.season.2 <- data.season.2[ ave(1:nrow(data.season.2), data.season.2$ringnr,
                                   FUN=length) > 1 , ]
data.season.2$ringnr <- droplevels(data.season.2$ringnr)
table(data.season.2$breeding_stage)
```

```
##
## Before After
##       8      8
```

N = 16 samples from 8 females sampled twice (8 during incubation and 8 during nestling rearing)

1.3.5 Concentration differences between groups

We have noted that samples vary in their concentration in preen oil (proxy: abundance of the most common substance), and that preen oil concentration affect chromatographic data, with low-concentration samples having less substances and lower diversity. Before testing for differences between groups (females vs males, incubation vs nestling rearing, pairs, individuals) in preen oil composition (dissimilarity matrix, richness, diversity, volatility), we should make sure that there is no difference between groups in concentration. If there is, it would mean that the difference in preen oil composition observed may simply be due to the concentration bias.

For sex and breeding stage differences in concentration, we use Wilcoxon rank sum tests. To test for differences in concentration among pairs and individuals, we use Spearman correlation tests. We use non-parametric tests because the assumptions of normality and homoscedasticity were not met.

Difference in concentration between sexes

```
# complete data set
wilcox.test(abundance_mostcommonpeak ~ sex, data = data.sex.1)
```

```
##
## Wilcoxon rank sum test
##
## data: abundance_mostcommonpeak by sex
## W = 117, p-value = 0.3571
## alternative hypothesis: true location shift is not equal to 0
```

```
# reduced data set
wilcox.test(abundance_mostcommonpeak ~ sex, data = data.sex.2)
```

```
##
## Wilcoxon rank sum test
##
## data: abundance_mostcommonpeak by sex
## W = 60, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

No difference between male and female samples (in both data sets with complete pairs)

Difference in concentration between breeding stages

```
# complete data set
wilcox.test(abundance_mostcommonpeak ~ breeding_stage, data = data.season.1)
```

```
##
## Wilcoxon rank sum test
##
## data: abundance_mostcommonpeak by breeding_stage
## W = 65, p-value = 0.7125
## alternative hypothesis: true location shift is not equal to 0
```

```
# reduced data set
wilcox.test(abundance_mostcommonpeak ~ breeding_stage, data = data.season.2)
```

```
##
## Wilcoxon rank sum test
##
## data: abundance_mostcommonpeak by breeding_stage
## W = 32, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

No difference between incubation and nestling rearing samples (in both data sets with females sampled twice)

Difference in concentration among pairs

```
# complete data set
# data about pairs and concentration
data.sex.1.conc.pairs <- data.sex.1[,c(2,8,17)]
# convert in horizontal format
data.sex.1.conc.pairs <- spread(data.sex.1.conc.pairs, sex, abundance_mostcommonpeak)
# correlation test
cor.test(data.sex.1.conc.pairs$Female,
         data.sex.1.conc.pairs$Male,
         method = "spearman")
```

```
##
## Spearman's rank correlation rho
##
## data: data.sex.1.conc.pairs$Female and data.sex.1.conc.pairs$Male
## S = 300, p-value = 0.007762
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.6323529
```

```

# reduced data set
# data about pairs and concentration
data.sex.2.conc.pairs <- data.sex.2[,c(2,8,17)]
# convert in horizontal format
data.sex.2.conc.pairs <- spread(data.sex.2.conc.pairs, sex, abundance_mostcommonpeak)
# correlation test
cor.test(data.sex.2.conc.pairs$Female,
          data.sex.2.conc.pairs$Male,
          method = "spearman")

```

```

##
## Spearman's rank correlation rho
##
## data: data.sex.2.conc.pairs$Female and data.sex.2.conc.pairs$Male
## S = 56, p-value = 0.01187
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.7454545

```

The concentrations in samples from males and females from the same pair are correlated

Difference in concentration among individuals

```

# complete data set
# data about individuals and concentration
data.season.1.conc.ind <- data.season.1[,c(3,7,17)]
# convert in horizontal format
data.season.1.conc.ind <- spread(data.season.1.conc.ind, breeding_stage,
                                abundance_mostcommonpeak)
# correlation test
cor.test(data.season.1.conc.ind$Before,
          data.season.1.conc.ind$After,
          method = "spearman")

```

```

##
## Spearman's rank correlation rho
##
## data: data.season.1.conc.ind$Before and data.season.1.conc.ind$After
## S = 338, p-value = 0.573
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.1818182

```

```

# reduced data set
# data about ind and concentration
data.season.2.conc.ind <- data.season.2[,c(3,7,17)]
# convert in horizontal format
data.season.2.conc.ind <- spread(data.season.2.conc.ind, breeding_stage,
                                abundance_mostcommonpeak)
# correlation test

```

```
cor.test(data.season.2.conc.ind$Before,
         data.season.2.conc.ind$After,
         method = "spearman")

##
## Spearman's rank correlation rho
##
## data: data.season.2.conc.ind$Before and data.season.2.conc.ind$After
## S = 122, p-value = 0.2675
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.452381
```

No correlation in the concentration of preen oil in the samples of the same individual

2 Analysis

First, we test for differences in the overall composition of preen oil with permutational multivariate analyses of variance (PERMANOVA) using PRIMER software. These analyses are not detailed in this document.

Second, we test for differences in richness (number of substances), diversity (Shannon index) and volatility (proportion of high-volatility substances and proportion of low-volatility substances) of preen oil. To do so, we run generalized linear mixed models (GLMM) with Gaussian distribution and identity link using the *lmer* function of the *lme4* package. For fixed effects, we look at the β estimates and their 95% confidence intervals. Fixed effects are significant ($\alpha = 5\%$) if the confidence interval does not span 0. In addition, we run the model with the *lmerTest* package to get a P-value. We use the *partR2* package to get the marginal R² of the fixed effects and the *rptR* package to get the repeatability (adjusted repeatability using the LMM method) of the random effects. *rptR* also provides a P-value (based on permutations) for the random effects.

2.1 Sex and pair analysis

We test for sex differences within pairs (during nestling rearing).

2.1.1 Richness (S)

Fit the model

```
glmm.sex.S <- lme4::lmer(S ~ sex + (1|nestbox_factor), data = data.sex.2)
summary(glmm.sex.S)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: S ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 171
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.54872 -0.55813 0.00133 0.46450 1.81218
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
## nestbox_factor (Intercept) 241.36  15.54
## Residual              98.01   9.90
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##           Estimate Std. Error t value
## (Intercept)  53.636      5.554   9.656
## sexMale      -2.727      4.221  -0.646
##
## Correlation of Fixed Effects:
##           (Intr)
## sexMale -0.380
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glm.ssex.S, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect  group      term      estimate std.error statistic conf.low conf.high
##   <chr>   <chr>    <chr>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1 fixed  <NA>      (Intercep~  53.6      5.55      9.66      43.4      64.6
## 2 fixed  <NA>      sexMale    -2.73     4.22     -0.646    -11.6      4.92
## 3 ran_pars nestbox_f~ sd__(Inte~  15.5      NA        NA         5.79     23.6
## 4 ran_pars Residual  sd__Obser~   9.90     NA        NA         5.50     14.3
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(S ~ sex + (1|nestbox_factor), data = data.sex.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: S ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 171
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.54872 -0.55813  0.00133  0.46450  1.81218
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
## nestbox_factor (Intercept) 241.36  15.54
## Residual              98.01   9.90
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##           Estimate Std. Error    df t value Pr(>|t|)
```



```
## (Intercept)  53.636      5.554 13.282   9.656 2.25e-07 ***
## sexMale      -2.727      4.221 10.000  -0.646   0.533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## sexMale -0.380
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glmm.sex.S, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2      CI_lower CI_upper nboot ndf
## 0.0057 0          0.0917  1000  2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

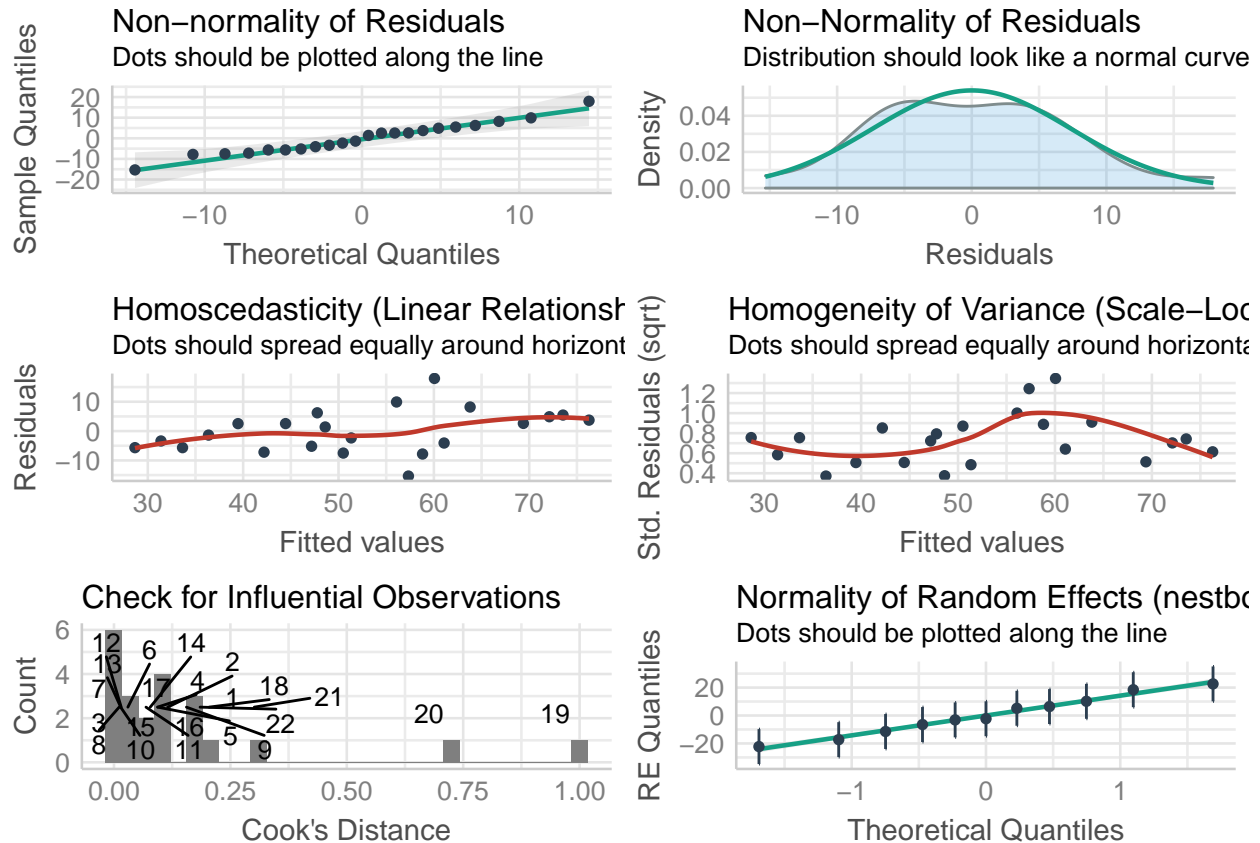
```
rpt(S ~ sex + (1|nestbox_factor),
    grname = "nestbox_factor",
    data = data.sex.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for nestbox_factor :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for nestbox_factor
## R   = 0.711
## SE  = 0.17
## CI  = [0.256, 0.913]
## P   = 0.00268 [LRT]
##      0.015 [Permutation]
```

Check the model assumptions

```
check_model(glmm.sex.S)
```



2.1.2 Diversity (H)

Fit the model

```
glmm.sex.H <- lme4::lmer(H ~ sex + (1|nestbox_factor), data = data.sex.2)
summary(glmm.sex.H)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: H ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: -21.7
##
## Scaled residuals:
##    Min      1Q  Median      3Q      Max
## -1.7324 -0.4671  0.2386  0.5233  1.4518
##
## Random effects:
## Groups          Name          Variance Std.Dev.
## nestbox_factor (Intercept) 0.014200 0.11917
## Residual              0.006868 0.08287
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##              Estimate Std. Error t value
```

```
## (Intercept)  3.15904    0.04376  72.183
## sexMale      -0.07825    0.03534  -2.214
##
## Correlation of Fixed Effects:
##      (Intr)
## sexMale -0.404
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glmmm.sex.H, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect    group      term      estimate std.error statistic conf.low conf.high
##   <chr>    <chr>    <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed    <NA>      (Intercep~  3.16     0.0438    72.2     3.07     3.25
## 2 fixed    <NA>      sexMale    -0.0782   0.0353   -2.21    -0.156   -0.0137
## 3 ran_pars nestbox_f~ sd__(Inte~  0.119    NA        NA        0.0490   0.191
## 4 ran_pars Residual  sd__Obser~  0.0829   NA        NA        0.0473   0.118
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(H ~ sex + (1|nestbox_factor), data = data.sex.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: H ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: -21.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.7324 -0.4671  0.2386  0.5233  1.4518
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## nestbox_factor (Intercept) 0.014200 0.11917
## Residual              0.006868 0.08287
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  3.15904    0.04376  13.75245  72.183  <2e-16 ***
## sexMale      -0.07825    0.03534  10.00000  -2.214  0.0512 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## sexMale -0.404
```

Get the marginal R² (with confidence interval) of the fixed effect

```
partR2(glm.m.sex.H, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2      CI_lower CI_upper nboot ndf
## 0.0707 0.0017    0.2747   1000  2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

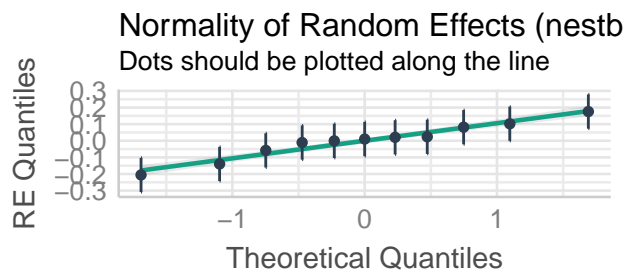
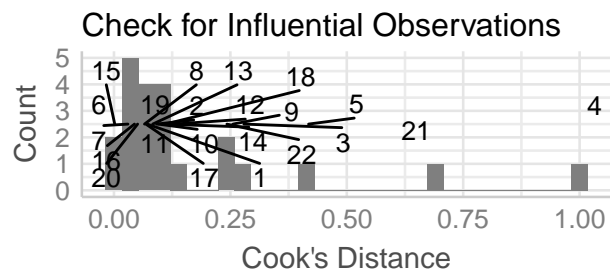
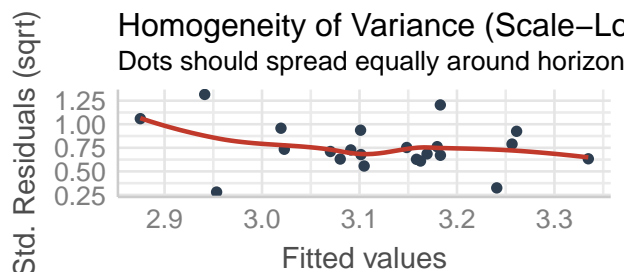
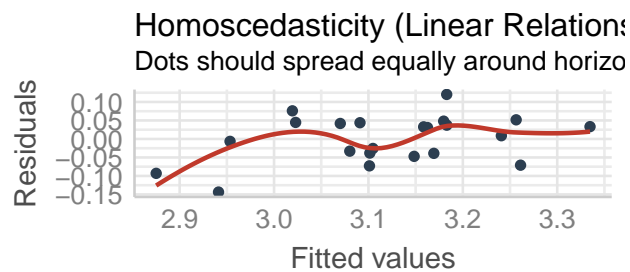
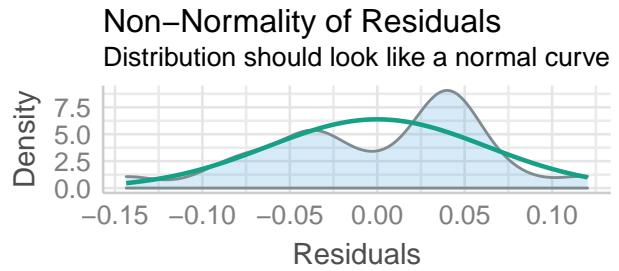
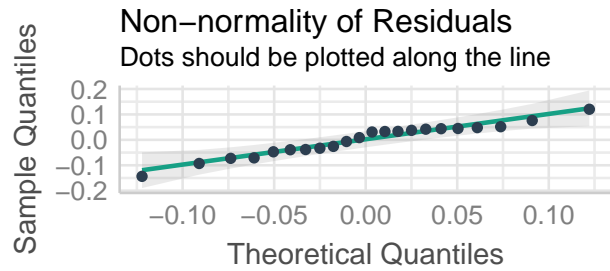
```
rpt(H ~ sex + (1|nestbox_factor),
    grname = "nestbox_factor",
    data = data.sex.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for nestbox_factor :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for nestbox_factor
## R   = 0.674
## SE  = 0.187
## CI  = [0.165, 0.908]
## P   = 0.00492 [LRT]
##      0.02 [Permutation]
```

Check the model assumptions

```
check_model(glm.m.sex.H)
```



Control for concentration

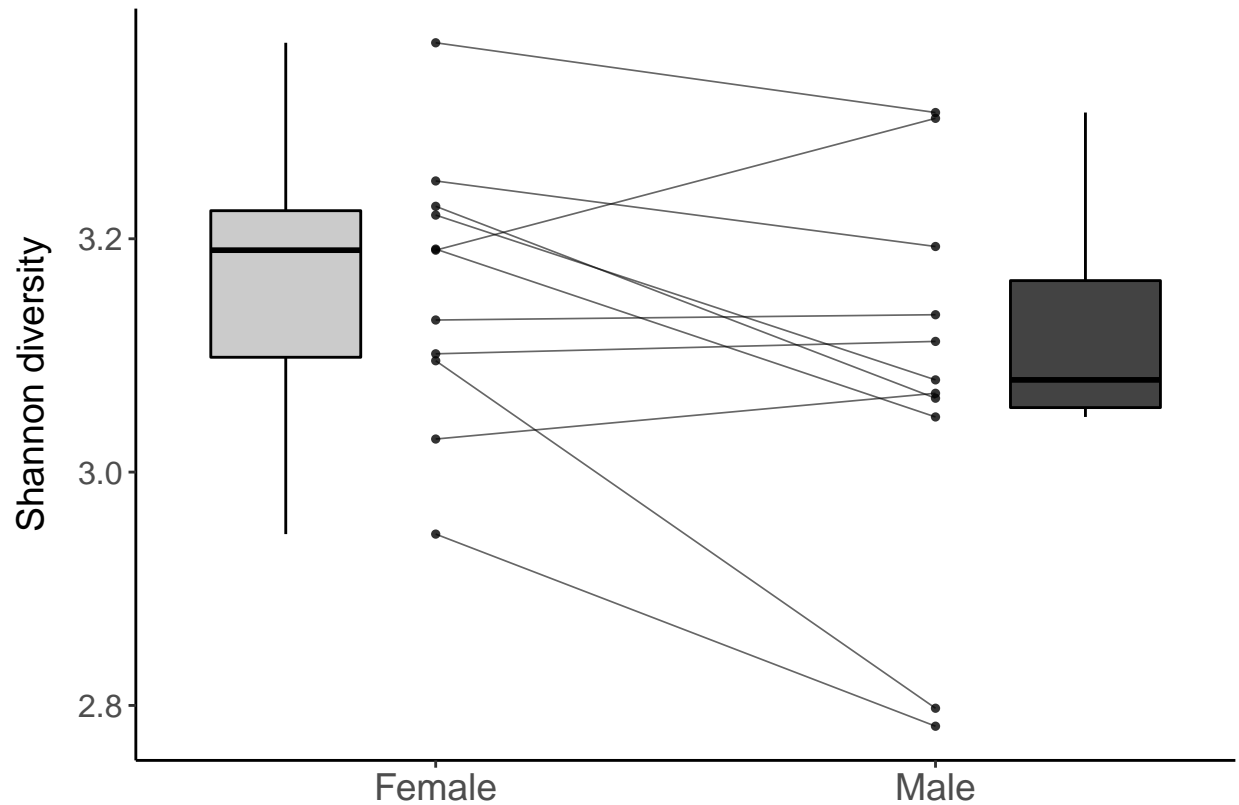
```
summary(lmerTest::lmer(H ~ abundance_mostcommonpeak + sex +
  (1|nestbox_factor), data = data.sex.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: H ~ abundance_mostcommonpeak + sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 8.2
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -1.8656 -0.4301  0.0569  0.3958  1.4193
##
## Random effects:
##   Groups                Name                Variance Std.Dev.
##   nestbox_factor (Intercept) 0.009798 0.09898
##   Residual                  0.007121 0.08438
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    3.096e+00  5.084e-02 1.638e+01  60.887  <2e-16 ***
## abundance_mostcommonpeak 4.483e-08  2.286e-08 1.873e+01   1.961   0.0649 .
```

```
## sexMale          -5.597e-02  3.773e-02  1.047e+01  -1.483   0.1674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) abndn_
## abndnc_mstc -0.636
## sexMale     -0.529  0.301
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

Boxplot of sex difference (within pairs) in diversity

```
theme_set(theme_classic())
plot.H.sex <- ggplot(data = data.sex.2, aes(x=sex, y=H)) +
  geom_point(data = data.sex.2, size = 1.3, shape = 16, alpha = .8,
            position = position_dodge(width = 0), show.legend = T,) +
  geom_line(aes(group=nestbox_factor), alpha = .6, size=0.3, position=position_dodge(0))+
  geom_boxplot(data = data.sex.2 %>% filter(sex == "Female"),
              aes(x=sex, y=H),
              position = position_nudge(x = -0.3), outlier.shape = NA, width = .3,
              alpha = .8, colour="black", fill = "grey") +
  geom_boxplot(data = data.sex.2 %>% filter(sex == "Male"),
              aes(x=sex, y=H),
              position = position_nudge(x = 0.3), outlier.shape = NA, width = .3,
              alpha = .8, colour="black", fill = "grey8") +
  scale_x_discrete(labels=c("Female", "Male")) +
  labs(x="", y="Shannon diversity") +
  theme(axis.title.y=element_text(size=14,
                                margin = margin(t = 0, r = 10, b = 0, l = 0)),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=14))
plot.H.sex
```



2.1.3 Proportion of high-volatility substances

Fit the model

```
glmm.sex.highvolatility <- lme4::lmer(prop.highvolatility ~ sex + (1|nestbox_factor),
                                     data = data.sex.2)
summary(glmm.sex.highvolatility)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: prop.highvolatility ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 73.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.98185 -0.25782 -0.04865  0.64868  1.70396
##
## Random effects:
## Groups       Name             Variance Std.Dev.
## nestbox_factor (Intercept)  0.000     0.000
## Residual                1.826     1.351
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
```

```
##           Estimate Std. Error t value
## (Intercept) 10.3071    0.4074 25.301
## sexMale     -1.3544    0.5761 -2.351
##
## Correlation of Fixed Effects:
##      (Intr)
## sexMale -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glmmm.sex.highvolatility, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term      estimate std.error statistic conf.low conf.high
##   <chr>   <chr>    <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed   <NA>      (Intercep~ 10.3      0.407     25.3      9.37     11.1
## 2 fixed   <NA>      sexMale    -1.35     0.576     -2.35     -2.46     -0.214
## 3 ran_pars nestbox_f~ sd__(Inte~ 0         NA        NA        0         1.15
## 4 ran_pars Residual  sd__Obser~ 1.35     NA        NA        0.769     1.65
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(prop.highvolatility ~ sex +
                        (1|nestbox_factor), data = data.sex.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: prop.highvolatility ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 73.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.98185 -0.25782 -0.04865  0.64868  1.70396
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## nestbox_factor (Intercept) 0.000    0.000
## Residual                1.826    1.351
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##           Estimate Std. Error    df t value Pr(>|t|)
## (Intercept) 10.3071    0.4074 20.0000 25.301 <2e-16 ***
## sexMale     -1.3544    0.5761 20.0000 -2.351 0.0291 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```



```
##          (Intr)
## sexMale -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glmm.sex.highvolatility, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2      CI_lower CI_upper nboot ndf
## 0.2083 0.0071    0.5593   1000  2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

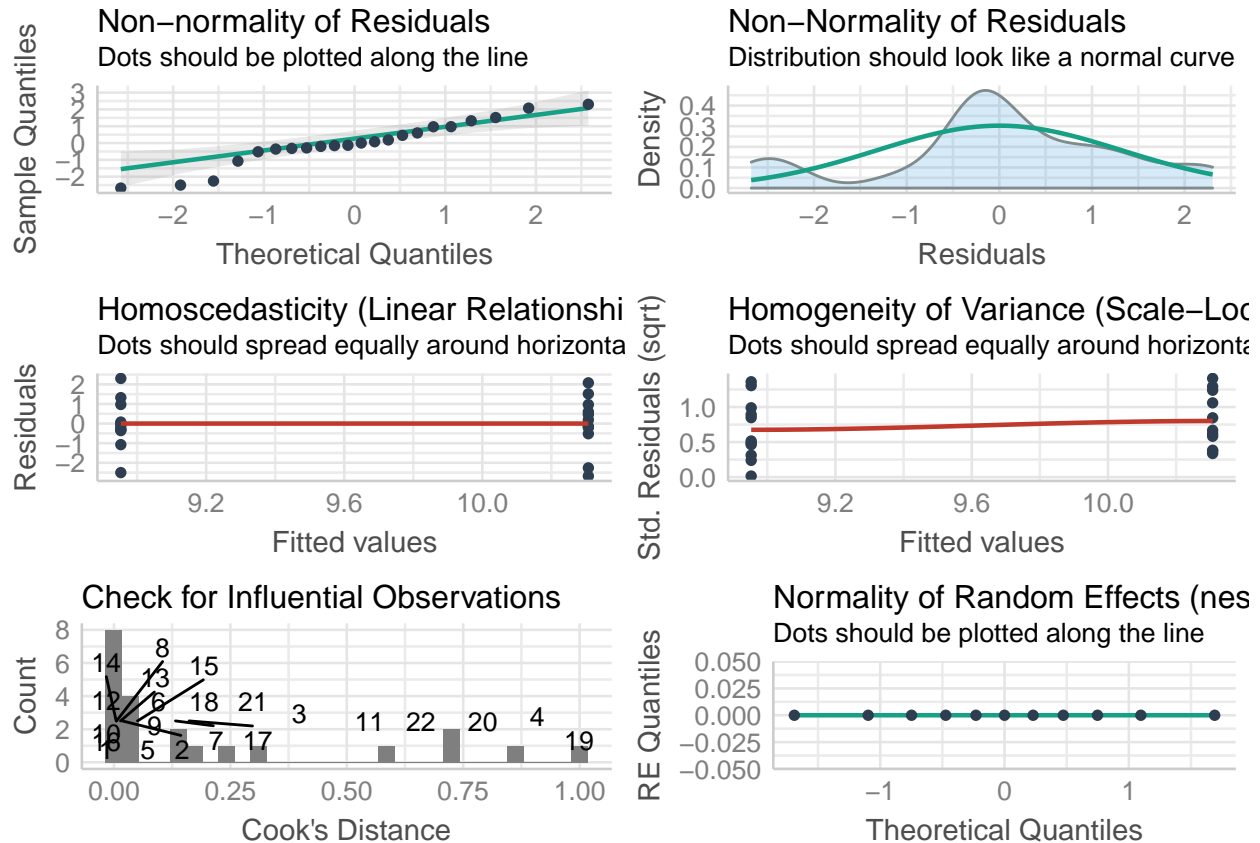
```
rpt(prop.highvolatility ~ sex + (1|nestbox_factor),
    grname = "nestbox_factor",
    data = data.sex.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for nestbox_factor :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for nestbox_factor
## R   = 0
## SE = 0.192
## CI = [0, 0.63]
## P   = 1 [LRT]
##      1 [Permutation]
```

Check the model assumptions

```
check_model(glmm.sex.highvolatility)
```



Control for concentration

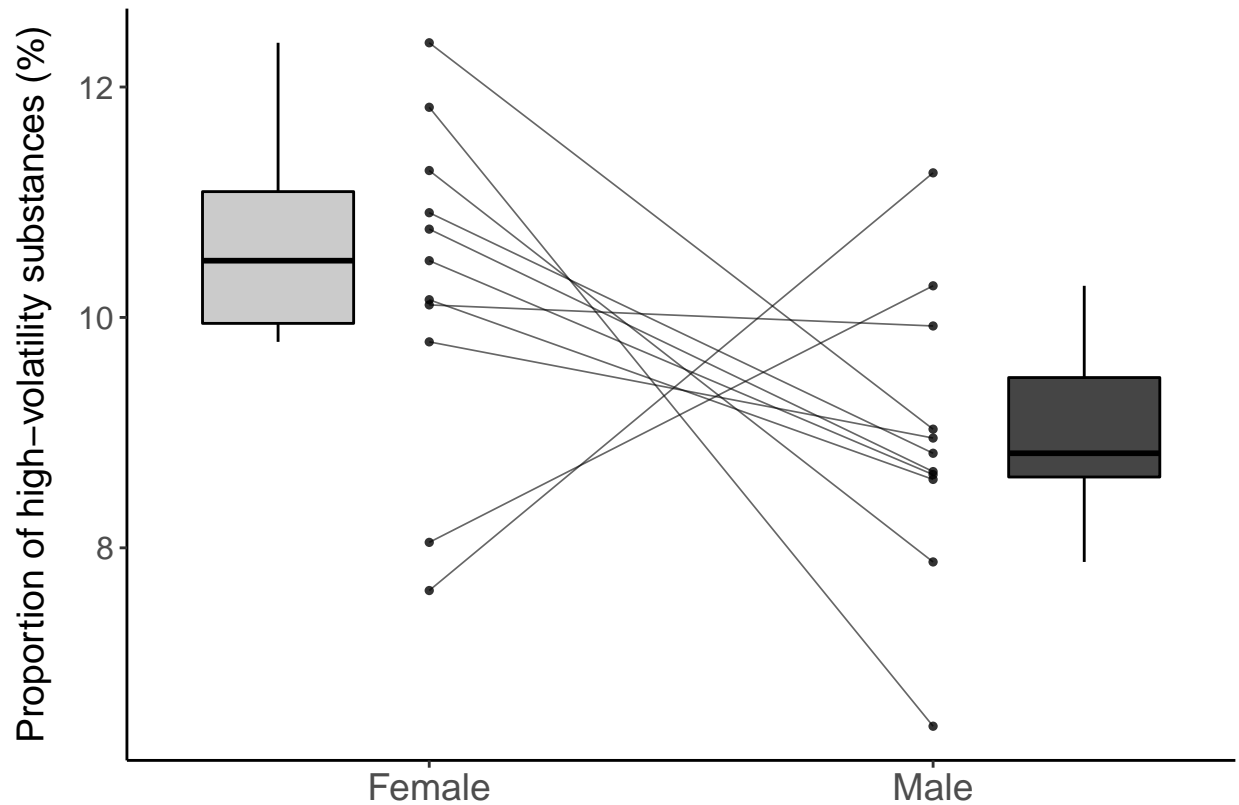
```
summary(lmerTest::lmer(prop.highvolatility ~ abundance_mostcommonpeak
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## prop.highvolatility ~ abundance_mostcommonpeak + sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 102.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.92861 -0.25130 -0.04577  0.63535  1.65986
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## nestbox_factor (Intercept) 0.000    0.000
## Residual          1.922    1.386
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   1.031e+01  5.351e-01  1.900e+01  19.27 6.26e-14 ***
## abundance_mostcommonpeak -2.460e-09  2.360e-07  1.900e+01  -0.01  0.9918
```

```
## sexMale          -1.356e+00  6.026e-01  1.900e+01   -2.25   0.0365 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) abndn_
## abndnc_mstc -0.624
## sexMale     -0.663  0.195
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Boxplot the sex difference (within pairs) in the proportion of high-volatility substances

```
theme_set(theme_classic())
plot.prophighvolatility.sex <- ggplot(data = data.sex.2, aes(x=sex, y=prop.highvolatility)) +
  geom_point(data = data.sex.2, size = 1.3, shape = 16, alpha = .8,
            position = position_dodge(width = 0), show.legend = T,) +
  geom_line(aes(group=nestbox_factor), alpha = .6, size=0.3, position=position_dodge(0))+
  geom_boxplot(data = data.sex.2 %>% filter(sex == "Female"),
              aes(x=sex, y=prop.highvolatility),
              position = position_nudge(x = -0.3), outlier.shape = NA, width = .3, alpha = .8,
              colour="black", fill="grey") +
  geom_boxplot(data = data.sex.2 %>% filter(sex == "Male"),
              aes(x=sex, y=prop.highvolatility),
              position = position_nudge(x = 0.3), outlier.shape = NA, width = .3, alpha = .8,
              colour="black", fill = "grey9") +
  scale_x_discrete(labels=c("Female", "Male")) +
  labs(x="", y="Proportion of high-volatility substances (%)") +
  theme(axis.title.y=element_text(size=14, margin = margin(t = 0, r = 10, b = 0, l = 0)),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=14))
plot.prophighvolatility.sex
```



2.1.4 Proportion of low-volatility substances

Fit the model

```
glmm.sex.lowvolatility <- lme4::lmer(prop.lowvolatility ~ sex + (1|nestbox_factor),
                                   data = data.sex.2)
summary(glmm.sex.lowvolatility)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: prop.lowvolatility ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 78
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.25894 -0.59950  0.05817  0.36284  1.53779
##
## Random effects:
## Groups       Name             Variance Std.Dev.
## nestbox_factor (Intercept) 1.607     1.268
## Residual                1.175     1.084
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
```

```
##           Estimate Std. Error t value
## (Intercept)  9.64528    0.50297  19.177
## sexMale      0.02959    0.46228   0.064
##
## Correlation of Fixed Effects:
##      (Intr)
## sexMale -0.460
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glmmm.sex.lowvolatility, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group      term      estimate std.error statistic conf.low conf.high
##   <chr>   <chr>    <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed   <NA>      (Intercep~  9.65     0.503    19.2     8.69    10.7
## 2 fixed   <NA>      sexMale     0.0296    0.462    0.0640  -0.887   0.884
## 3 ran_pars nestbox_f~ sd__(Inte~  1.27     NA       NA       0.146    2.00
## 4 ran_pars Residual  sd__Obser~  1.08     NA       NA       0.644    1.51
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(prop.lowvolatility ~ sex + (1|nestbox_factor),
                        data = data.sex.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: prop.lowvolatility ~ sex + (1 | nestbox_factor)
## Data: data.sex.2
##
## REML criterion at convergence: 78
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.25894 -0.59950  0.05817  0.36284  1.53779
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## nestbox_factor (Intercept) 1.607    1.268
## Residual              1.175    1.084
## Number of obs: 22, groups: nestbox_factor, 11
##
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  9.64528    0.50297 14.99644  19.177 5.82e-12 ***
## sexMale      0.02959    0.46228 10.00000   0.064    0.95
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## sexMale -0.460
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glmm.sex.lowvolatility, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2    CI_lower CI_upper nboot ndf
## 1e-04 0        0.1193   1000  2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

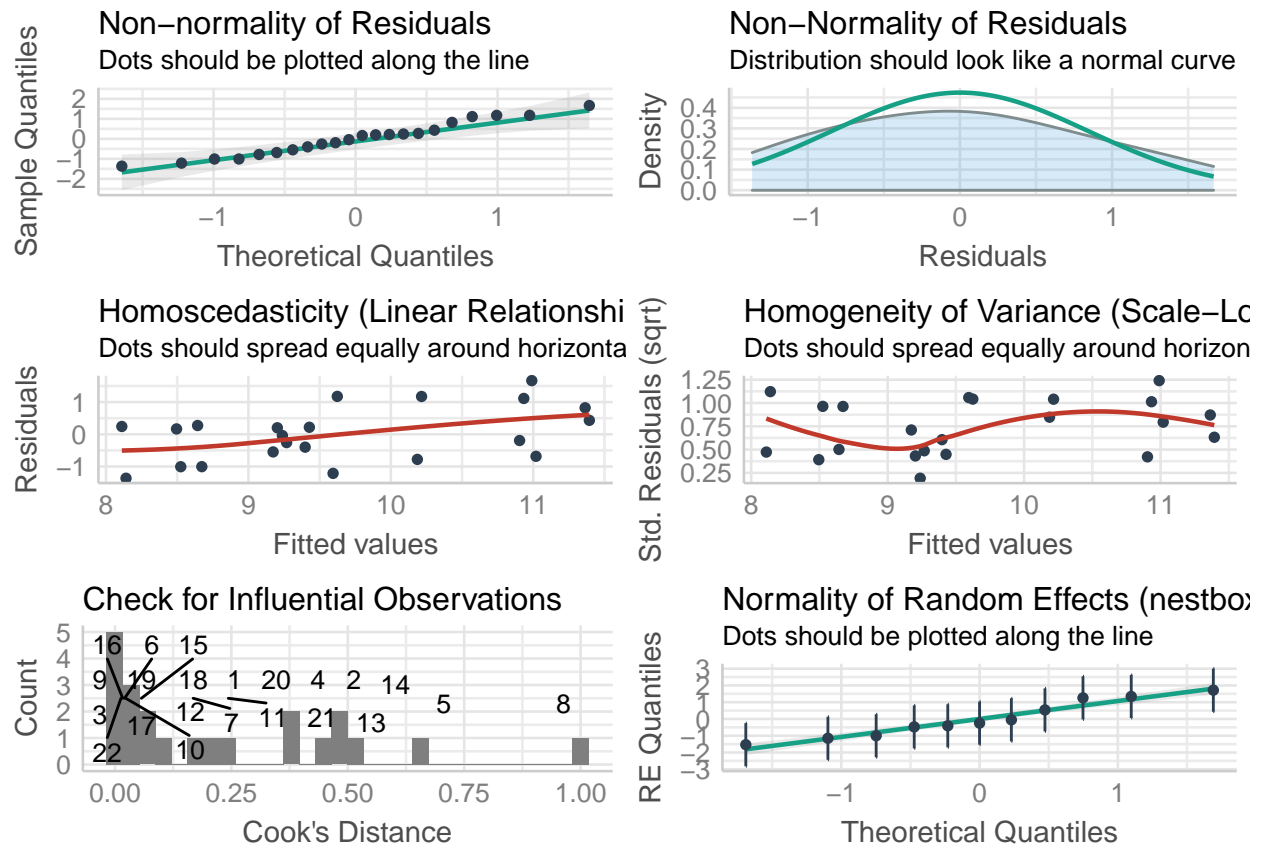
```
rpt(prop.lowvolatility ~ sex + (1|nestbox_factor),
    grname = "nestbox_factor",
    data = data.sex.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for nestbox_factor :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for nestbox_factor
## R   = 0.578
## SE  = 0.202
## CI  = [0.105, 0.864]
## P   = 0.0173 [LRT]
##     = 0.034 [Permutation]
```

Check the model assumptions

```
check_model(glm~sex~lowvolatility)
```



2.2 Season and pair analysis

We test for seasonal differences (incubation *vs* nestling rearing) within individual females.

2.2.1 Richness (S)

Fit the model

```
glmm.season.S <- lme4::lmer(S ~ breeding_stage + (1|ringnr), data = data.season.2)
summary(glmm.season.S)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: S ~ breeding_stage + (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: 128.2
##
## Scaled residuals:
##    Min      1Q  Median      3Q      Max
## -1.1649 -0.8645 -0.2774  0.8136  1.6950
##
```

```
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   ringnr   (Intercept)  0.0      0.00
##   Residual                411.3    20.28
## Number of obs: 16, groups:  ringnr, 8
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      44.500      7.170   6.206
## breeding_stageAfter  1.125     10.140   0.111
##
## Correlation of Fixed Effects:
##              (Intr)
## brdng_stgAf -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glmmm.season.S, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group   term          estimate std.error statistic conf.low conf.high
##   <chr>   <chr>   <chr>          <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1 fixed   <NA>   (Intercept)    44.5       7.17      6.21      30.8      58.1
## 2 fixed   <NA>   breeding_sta~  1.12      10.1      0.111    -19.9      21.5
## 3 ran_pars ringnr sd__(Interce~  0         NA        NA        0         19.3
## 4 ran_pars Residu~ sd__Observat~ 20.3       NA        NA       10.1      27.0
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(S ~ breeding_stage + (1|ringnr), data = data.season.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: S ~ breeding_stage + (1 | ringnr)
##   Data: data.season.2
##
## REML criterion at convergence: 128.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.1649 -0.8645 -0.2774  0.8136  1.6950
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   ringnr   (Intercept)  0.0      0.00
##   Residual                411.3    20.28
## Number of obs: 16, groups:  ringnr, 8
##
## Fixed effects:
##               Estimate Std. Error    df t value Pr(>|t|)
```



```
## (Intercept)          44.500      7.170 14.000   6.206 2.29e-05 ***
## breeding_stageAfter    1.125     10.140 14.000   0.111   0.913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## brdng_stgAf -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glmm.season.S, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2    CI_lower CI_upper nboot ndf
## 8e-04 1e-04    0.2908   1000  2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability of the random effect

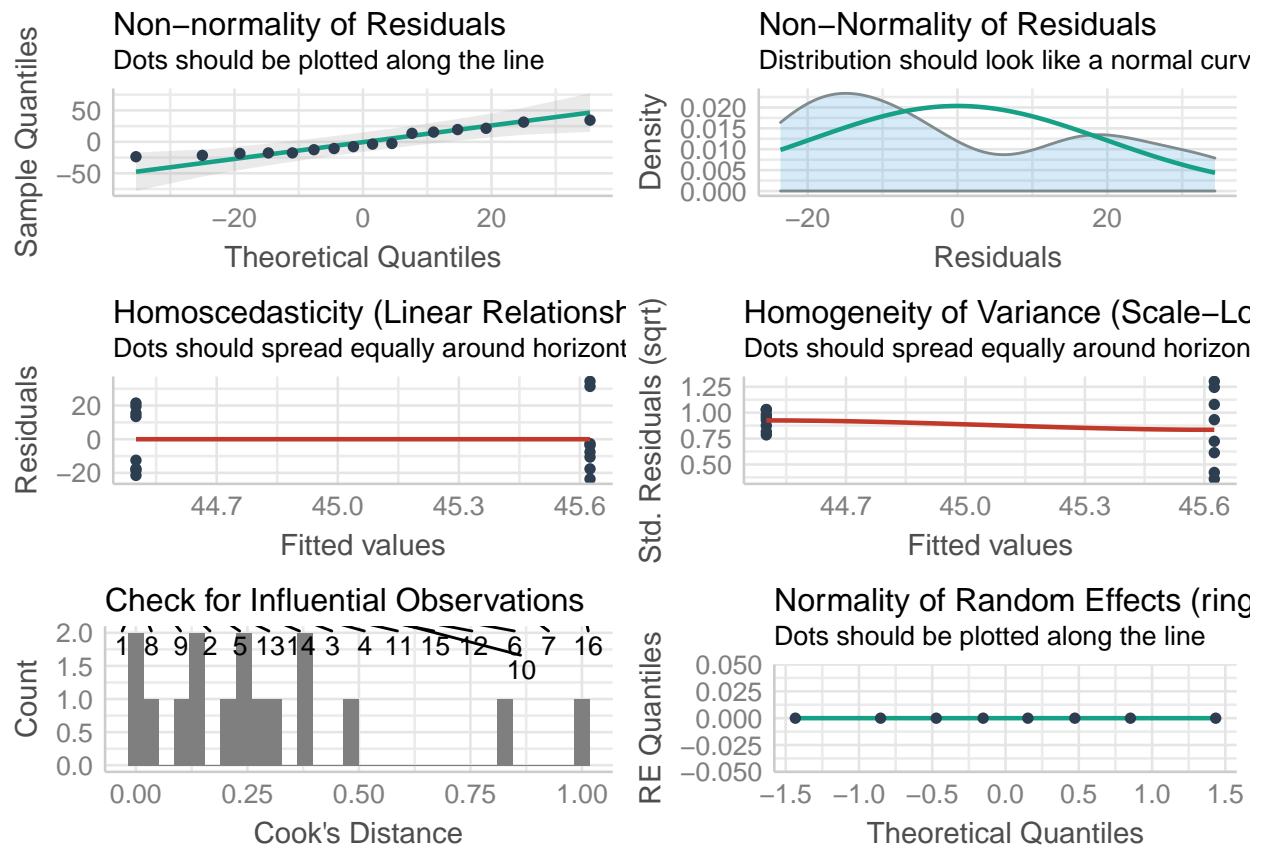
```
rpt(S ~ breeding_stage + (1|ringnr),
    grname = "ringnr",
    data = data.season.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for ringnr :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for ringnr
## R    = 0
## SE = 0.217
## CI = [0, 0.705]
## P    = 1 [LRT]
##      1 [Permutation]
```

Check the model assumptions

```
check_model(glm season.S)
```



2.2.2 Diversity (H)

Fit the model

```
glm season.H <- lme4::lmer(H ~ breeding_stage + (1|ringnr), data = data season.2)
summary(glm season.H)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: H ~ breeding_stage + (1 | ringnr)
## Data: data season.2
##
## REML criterion at convergence: -5.1
##
## Scaled residuals:
##    Min      1Q  Median      3Q      Max
## -1.6426 -0.7893  0.1148  0.8722  1.6648
##
## Random effects:
## Groups Name Variance Std.Dev.
## ringnr (Intercept) 0.00000 0.0000
## Residual 0.03024 0.1739
## Number of obs: 16, groups: ringnr, 8
```

```
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      3.02601    0.06149  49.215
## breeding_stageAfter 0.05246    0.08695   0.603
##
## Correlation of Fixed Effects:
##           (Intr)
## brdng_stgAf -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glm.lmm.season.H, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect  group term          estimate std.error statistic conf.low conf.high
##   <chr>   <chr> <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>
## 1 fixed   <NA> (Intercept)    3.03      0.0615     49.2      2.90      3.14
## 2 fixed   <NA> breeding_sta~ 0.0525     0.0870     0.603    -0.113     0.232
## 3 ran_pars ringnr sd__(Interce~ 0          NA         NA         0         0.170
## 4 ran_pars Residu~ sd__Observat~ 0.174      NA         NA         0.0841     0.234
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(H ~ breeding_stage + (1|ringnr), data = data.season.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: H ~ breeding_stage + (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: -5.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.6426 -0.7893  0.1148  0.8722  1.6648
##
## Random effects:
## Groups Name Variance Std.Dev.
## ringnr (Intercept) 0.00000 0.0000
## Residual 0.03024 0.1739
## Number of obs: 16, groups: ringnr, 8
##
## Fixed effects:
##               Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)      3.02601    0.06149 14.00000  49.215 <2e-16 ***
## breeding_stageAfter 0.05246    0.08695 14.00000   0.603  0.556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##          (Intr)
## brdng_stgAf -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glmm.season.H, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2      CI_lower CI_upper nboot ndf
## 0.0237 1e-04    0.3318   1000   2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

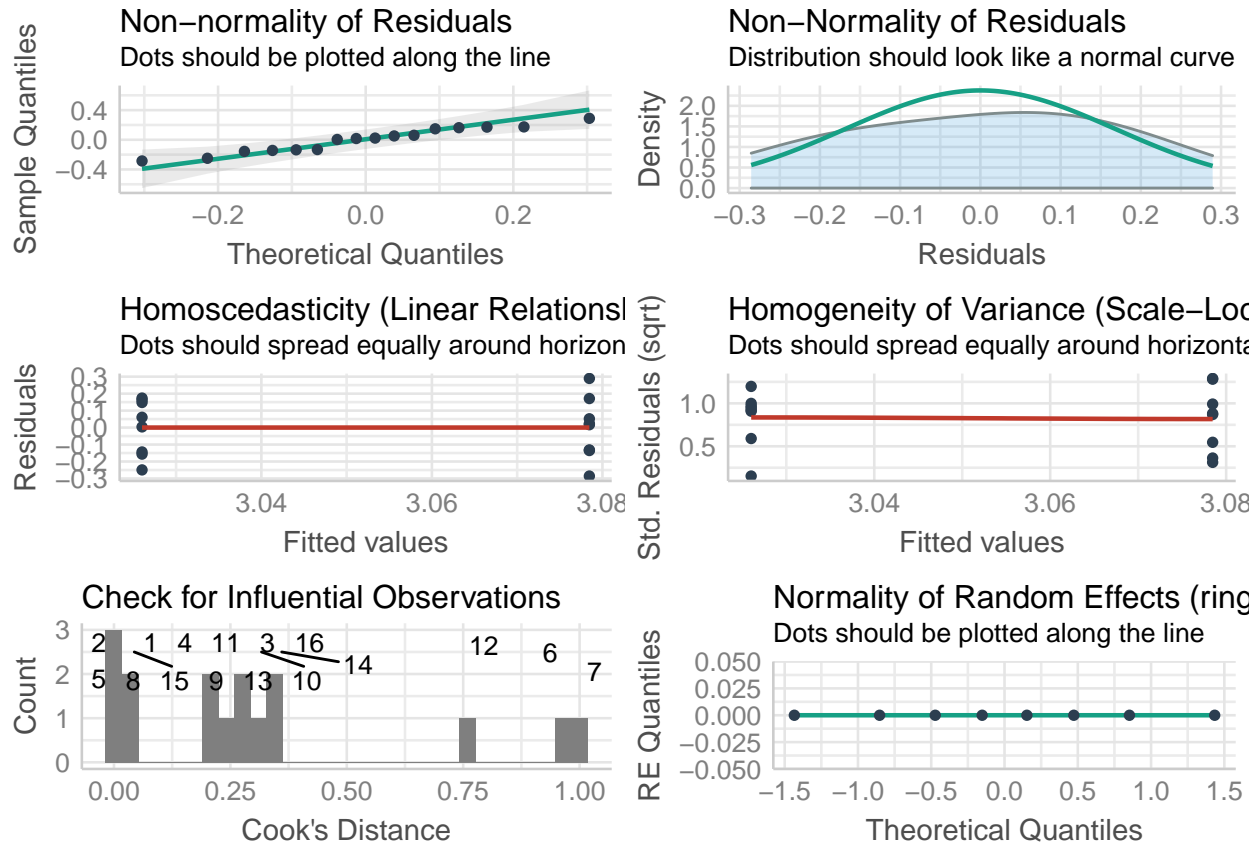
```
rpt(H ~ breeding_stage + (1|ringnr),
    grname = "ringnr",
    data = data.season.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for ringnr :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for ringnr
## R   = 0
## SE = 0.221
## CI = [0, 0.711]
## P   = 1 [LRT]
##      1 [Permutation]
```

Check model assumptions

```
check_model(glmm.season.H)
```



2.2.3 Proportion of high-volatility substances

Fit the model

```
glmm.season.highvolatility <- lme4::lmer(prop.highvolatility ~ breeding_stage +
                                         (1|ringnr), data = data.season.2)
summary(glmm.season.highvolatility)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: prop.highvolatility ~ breeding_stage + (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: 64.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.55311 -0.35630 -0.05179  0.49134  1.27176
##
## Random effects:
## Groups Name Variance Std.Dev.
## ringnr (Intercept) 2.415 1.554
## Residual 2.496 1.580
## Number of obs: 16, groups: ringnr, 8
##
## Fixed effects:
```

```
##               Estimate Std. Error t value
## (Intercept)      11.5257    0.7836  14.710
## breeding_stageAfter -1.1185    0.7900  -1.416
##
## Correlation of Fixed Effects:
##           (Intr)
## brdng_stgAf -0.504
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glmm.season.highvolatility, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect   group term          estimate std.error statistic conf.low conf.high
##   <chr>   <chr> <chr>          <dbl>     <dbl>     <dbl>   <dbl>   <dbl>
## 1 fixed   <NA> (Intercept)    11.5      0.784     14.7    10.1    13.1
## 2 fixed   <NA> breeding_sta~ -1.12     0.790     -1.42   -2.85    0.414
## 3 ran_pars ringnr sd__(Interce~  1.55     NA        NA       0       2.71
## 4 ran_pars Residu~ sd__Observat~  1.58     NA        NA      0.763    2.28
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(prop.highvolatility ~ breeding_stage + (1|ringnr),
                        data = data.season.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: prop.highvolatility ~ breeding_stage + (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: 64.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.55311 -0.35630 -0.05179  0.49134  1.27176
##
## Random effects:
## Groups Name Variance Std.Dev.
## ringnr (Intercept) 2.415 1.554
## Residual 2.496 1.580
## Number of obs: 16, groups: ringnr, 8
##
## Fixed effects:
##               Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)      11.5257    0.7836 11.2735  14.710 1.04e-08 ***
## breeding_stageAfter -1.1185    0.7900  7.0000  -1.416    0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## brdng_stgAf -0.504
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glm season.highvolatility, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
## R2      CI_lower CI_upper nboot ndf
## 0.0636 2e-04    0.3459   1000   2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

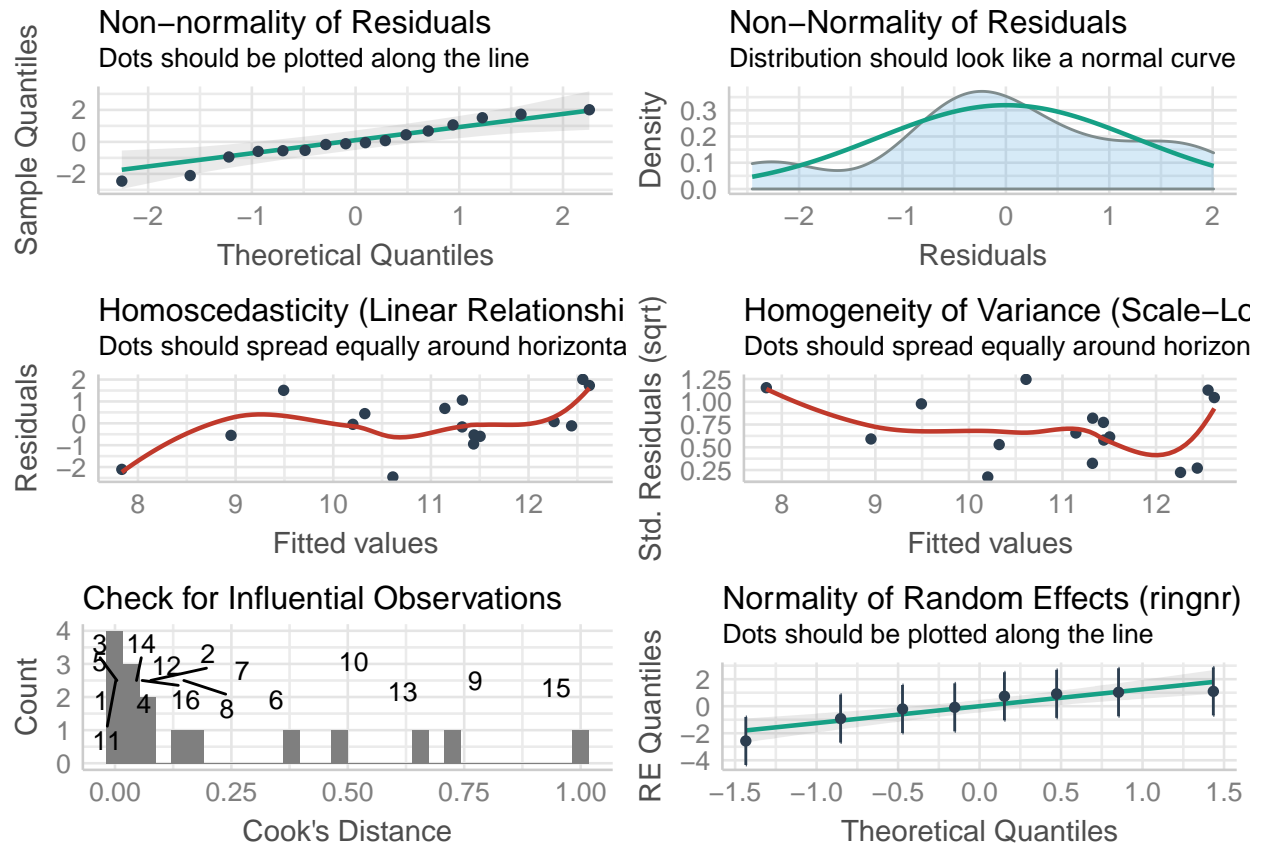
```
rpt(prop.highvolatility ~ breeding_stage + (1|ringnr),
    grname = "ringnr",
    data = data.season.2, datatype = "Gaussian",
    nboot = 1000, npermut = 1000,
    adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for ringnr :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for ringnr
## R  = 0.492
## SE = 0.254
## CI = [0, 0.882]
## P  = 0.0683 [LRT]
##    0.126 [Permutation]
```

Check the model assumptions

```
check_model(glm season.highvolatility)
```



2.2.4 Proportion of low-volatility substances

Fit the model

```
glm season.lowvolatility <- lme4::lmer(prop.lowvolatility ~
  breeding_stage + (1|ringnr),
  data = data.season.2)
summary(glm season.lowvolatility)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: prop.lowvolatility ~ breeding_stage + (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: 46.2
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -1.5382 -0.5605 -0.1636  0.2135  2.6798
##
## Random effects:
## Groups Name Variance Std.Dev.
## ringnr (Intercept) 0.00 0.000
```



```
## Residual          1.18      1.086
## Number of obs: 16, groups: ringnr, 8
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      7.4342    0.3840 19.360
## breeding_stageAfter 1.8427    0.5431  3.393
##
## Correlation of Fixed Effects:
##              (Intr)
## brdng_stgAf -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the beta estimate (with confidence interval) of the fixed effect

```
tidy(glmmm.season.lowvolatility, conf.int = TRUE, conf.method = 'boot')
```

```
## # A tibble: 4 x 8
##   effect  group  term          estimate std.error statistic conf.low conf.high
##   <chr>   <chr>  <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>
## 1 fixed  <NA>   (Intercept)      7.43      0.384      19.4      6.69      8.24
## 2 fixed  <NA>   breeding_sta~    1.84      0.543       3.39     0.748     2.88
## 3 ran_pars ringnr sd__(Interce~    0         NA         NA         0         1.01
## 4 ran_pars Residu~ sd__Observat~    1.09      NA         NA         0.454     1.41
```

Get the P-value of the fixed effect

```
summary(lmerTest::lmer(prop.lowvolatility ~ breeding_stage + (1|ringnr),
  data = data.season.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: prop.lowvolatility ~ breeding_stage + (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: 46.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.5382 -0.5605 -0.1636  0.2135  2.6798
##
## Random effects:
## Groups Name Variance Std.Dev.
## ringnr (Intercept) 0.00 0.000
## Residual 1.18 1.086
## Number of obs: 16, groups: ringnr, 8
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)      7.4342    0.3840 14.0000 19.360 1.67e-11 ***
## breeding_stageAfter 1.8427    0.5431 14.0000  3.393 0.00437 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## brdng_stgAf -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Get the marginal R^2 (with confidence interval) of the fixed effect

```
partR2(glmm.season.lowvolatility, nboot = 1000)
```

```
##
##
## R2 (marginal) and 95% CI for the full model:
##   R2      CI_lower CI_upper nboot ndf
## 0.4343 0.1156    0.7206   1000   2
##
## -----
##
## Part (semi-partial) R2:
## [1] "No partitions selected."
```

Get the repeatability (with confidence interval) of the random effect

```
rpt(prop.lowvolatility ~ breeding_stage + (1|ringnr),
     grname = "ringnr",
     data = data.season.2, datatype = "Gaussian",
     nboot = 1000, npermut = 1000,
     adjusted = TRUE)
```

```
## Bootstrap Progress:
## Permutation Progress for ringnr :

##
##
## Repeatability estimation using the lmm method
##
## Repeatability for ringnr
## R   = 0
## SE  = 0.22
## CI  = [0, 0.691]
## P   = 0.5 [LRT]
##     1 [Permutation]
```

Control for concentration

```
summary(lmerTest::lmer(prop.lowvolatility ~ abundance_mostcommonpeak +
                        breeding_stage + (1|ringnr),
                        data = data.season.2))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: prop.lowvolatility ~ abundance_mostcommonpeak + breeding_stage +
## (1 | ringnr)
## Data: data.season.2
##
## REML criterion at convergence: 73.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4405 -0.4679 -0.1114  0.3677  2.4800
##
## Random effects:
## Groups   Name            Variance Std.Dev.
## ringnr   (Intercept)  0.000      0.000
## Residual                1.119      1.058
## Number of obs: 16, groups: ringnr, 8
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    7.246e+00  4.002e-01 1.300e+01  18.104 1.33e-10 ***
## abundance_mostcommonpeak 2.819e-07  2.130e-07 1.300e+01   1.324  0.20836
## breeding_stageAfter    1.713e+00  5.380e-01 1.300e+01   3.185  0.00717 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) abndn_
## abndnc_mstc -0.356
## brdng_stgAf -0.585 -0.182
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Boxplot the change across breeding stages (within individual females) in the proportion of low-volatility substances

```
theme_set(theme_classic())
plot.proplowvolatility.season <- ggplot(data = data.season.2, aes(x=breeding_stage, y=prop.lowvolatility)) +
  geom_point(data = data.season.2, size = 1.3, shape = 16, alpha = .8, position = position_dodge(width = 1)) +
  geom_line(aes(group=ringnr), alpha = .6, size=0.3, position=position_dodge(0)) +
  geom_boxplot(data = data.season.2 %>% filter(breeding_stage == "Before"),
    aes(x=breeding_stage, y=prop.lowvolatility),
    position = position_nudge(x = -0.3), outlier.shape = NA, width = .3, alpha = .8,
    colour="black", fill="#66cdff") +
  geom_boxplot(data = data.season.2 %>% filter(breeding_stage == "After"),
    aes(x=breeding_stage, y=prop.lowvolatility),
    position = position_nudge(x = 0.3), outlier.shape = NA, width = .3, alpha = .8,
    colour="black", fill = "#97704d") +
  scale_x_discrete(labels=c("Incubation", "Nestling rearing")) +
  labs(x="", y="Proportion of low-volatility substances (%)") +
  theme(axis.title.y=element_text(size=14, margin = margin(t = 0, r = 10, b = 0, l = 0)),
    axis.text.y = element_text(size=12),
```

```
axis.text.x = element_text(size=14))  
plot.propvolvolatility.season
```

