Supplement to Cusson et al. (2025) – Mar Ecol Prog Ser 755: 133–150 – https://doi.org/10.3354/meps14790

Supplementary file: R code

Cusson, P.O., Bordeleau, X., Hammill, M.O. & Pelletier, F.

Table of contents

Packages

```
## Load required libraries
# Data manip
library(tibble) # Simple data frame
library(dplyr) # Manipulate data
library(tidyr) # Tidy data
library(hablar) # Convert column types
library(forcats) # Dealing with factors
library(lubridate) # Dealing with dates
library(stringr) # Dealing with strings
# Spatial
library(aniMotum) # State-space model for quality control of location data
library(move) # Dynamic Brownian bridge movement model (dbbmm)
library(adehabitatHR) # Calculate home range from dbbmm utilization distribution
library(sf) # Dealing with spatial features
library(lwgeom) # Tools to extract info from spatial features (ex: home range polygon)
library(rnaturalearth) # World map
library(rnaturalearthdata) # World map data
# Graphics
library(ggplot2) # Base for plots
library(scales) # Custom plot axis
library(ggridges) # Enables ridgeline plots
library(ggpubr) # Publication ready themes for ggplot
library(cowplot) # Combine plots in multi-panel figure
library(ggspatial) # Interacting with spatial data using ggplot2
library(ggsflabel) # Labels for sf objects in ggplot
library(ggrepel) # Text and label geoms for ggplot
# Statistical models
library(brms) # Bayesian regression models in Stan
library(BayesPostEst) # Tools for brms models
library(bayesplot) # Tools to plot brms models
library(tidybayes) # Tools for posterior distributions
# Tables
library(flextable) # Generate and customize tables
library(officer) # Save tables as word documents
```

Import “raw” locations

```
## Import raw locations from GSL tags and reformat the data for the ssm
RawLoc <-
  # Import raw locations
  readRDS(file = "Data/RawLoc_Meta.RDS") %>%
  # Select and rename desired columns for the 'fit_ssm' function
  dplyr::select(id, date, lc = class, lon = longitude,
                lat =latitude, smaj, smin, eor) %>%
  as_tibble(.name_repair = "minimal") %>%
  distinct(id, date, .keep_all = TRUE)
```

State-space model

```
# Generate empty tibble to store the fit_ssm output
ssm_fit <- tibble(id = character(),
                  ssm = list(),
                  converged = logical(),
                  pdHess = logical(),
                  pmodel = character())

# Give proper class to the tibble
class(ssm_fit) <- c("ssm_df", "tbl_df", "tbl", "data.frame")

for (i in unique(RawLoc$id)) {

  # Fit a state-space model for one individual
  ssm_id <- fit_ssm(
    filter(RawLoc, id == i), #data
    vmax = 5, #max travel rate (m/s) passed to pre-filter,to identify outlier locations
    ang = c(15, 25), #angles (deg) of outlier location "spikes"
    distlim = c(2500, 5000), #lengths (m) of outlier location "spikes"
    spdf = TRUE, #turn trip::sda on
    min.dt = 60, #minimum allowable time difference between observations in seconds
    pf = FALSE, #pre-filter the data and fit the SSM
    model = "crw", #fit a correlated random walk ("crw")
    time.step = NA, #locations are only estimated at observation times
    control =
      ssm_control(se = FALSE) # Don't calculate standard error for movement parameters to speed up calculation time
    )

  # Bind output for one individual to output from other models
  ssm_fit <- bind_rows(ssm_fit, ssm_id)

  # Remove ssm_id from the environment to avoid overwriting
  rm(ssm_id)

  # Clear memory using garbage collection to avoid RAM issue
  gc()
}

# Grab fitted values as an un-projected tibble
fit_loc <- grab(ssm_fit,
                what = "fitted",
                as_sf = FALSE)
```

Location error

```
## Location error
# Calculate single value of error per location
fit_loc <- fit_loc %>%
  mutate(x.se= x.se*1000, # convert from km to m
         y.se= y.se*1000, # convert from km to m
         # calculate the radius of a circle of equivalent area to the error ellipse
         radius = sqrt(x.se*y.se))
```

Home range estimation

Estimate monthly utilization distribution and calculate 95% home ranges using dynamic Brownian bridge movement models.

Format data

```
# Import Metadata
Metadata <- readRDS(file = "Data/Metadata.rds")

# Add deployment year column to fitted locations
fit_loc_hr <- left_join(fit_loc,
                        dplyr::select(Metadata, id, deploy.yr = year),
                        by=c("id"))
```

Function

```
# Write function to estimate utilization distributions and calculate 95% home ranges
calculate_hr <-  function(fit.yr) {

    # Generate empty tibble to store results of function
    fit.month.hr.temp <- tibble(id = character(), # seal id
                                deploy.yr = double(), # deployment year
                                month = numeric(), # month in which home range occurred
                                year = numeric(), # year in which home range occurred
                                dbbmm = list(), # dynamic Brownian bridge movement model
                                ext = numeric(), # extent parameter
                                poly95 = list(), # 95% home range polygon
                                ud = list(), # utilization distribution
                                prj = list(), # projected locations
                                move = list() # move object
    )

    for (i in unique(fit.yr$id)) {
```

```
# Filter data for one id
fit.month.id.data <- filter(fit.yr, id == i) %>%
  # Add month and day column
  mutate(month = month(date),
         day = day(date)) %>%
  # Group data by month
  group_by(month) %>%
  # Add column with number of days with data per month
  mutate(nb_days = length(unique(day))) %>%
  # Remove months with less then 9 days of data or 5 days in January because
  # transmission is paused during prolonged haulout during reproduction period
  filter(nb_days >= 9 | (month == 1 & nb_days >=5)) %>%
  # Ungroup data
  ungroup()

for (n in unique(month(fit.month.id.data$date))) {

  skip_to_next <- FALSE

  # Select locations for nth month
  fit.month.data <- fit.month.id.data[month(fit.month.id.data$date) == n,]

  # Generate move object
  fit.month.move <- move(x = fit.month.data$lon,
                         y = fit.month.data$lat,
                         time = as.POSIXct(fit.month.data$date,
                                           format="%Y-%m-%d %H:%M:%OS", tz="UTC"),
                         proj = CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"),
                         animal = fit.month.data$id)

  # Set projection of move object to flat coordinate system
  fit.month.prj <- spTransform(fit.month.move, center=TRUE)

  # calculate the dynamic Brownian motion variance of the gappy track
  fit.month.dbbv <- brownian.motion.variance.dyn(
    fit.month.prj,
    location.error = fit.month.data$radius,
    window.size = 7,
    margin = 3)

  # Ignore all segments that have a larger time lag than 12 hours.
  fit.month.dbbv@interest[timeLag(fit.month.prj,"hours")>=12] <- FALSE

  # then we use the 'dBMvariance' object to calculate the dBBMM
  ext <- 0.5
  while(TRUE) {
    tryCatch (
      fit.month.dbbmm <- suppressWarnings(brownian.bridge.dyn(
        fit.month.dbbv,
        raster = 1000,
        location.error = fit.month.data$radius,
        ext = ext,
        time.steps = 15,
        bbox = NA)),
      error = function(e) {
        message(paste("An error occurred in dbbmm for", i, n, "with ext =", ext, ":\n"), e)
        ext <<- ext + 0.5},
      finally = {
        if(exists("fit.month.dbbmm")) break
      }
    )
  }

  if(skip_to_next) { next }

  # Convert the raster produced by the dBBMM into a true utilization distribution (UD)
```

```r
        fit.month.dbbmm.sp <- as(fit.month.dbbmm, "SpatialPixelsDataFrame")
        fit.month.dbbmm.sp.ud <- new("estUD", fit.month.dbbmm.sp)
        fit.month.dbbmm.sp.ud@vol <- FALSE
        fit.month.dbbmm.sp.ud@h$meth <- "dBBMM"
        fit.month.dbbmm.ud <- suppressWarnings(getvolumeUD(fit.month.dbbmm.sp.ud, standardize = TRUE))

        # Pull out 95% contours
        tryCatch (

          fit.month.poly.95 <- suppressWarnings(getverticeshr(fit.month.dbbmm.ud, percent = 95)),

          error = function(e) {
            message(paste("An error occurred in hr 0.95 for", i, n,":\n"), e)
            skip_to_next <<- TRUE})

        if(skip_to_next) { next }

        # Store parts of the dbbmm and resulting UD and home range
        df.hr <- tibble(id = i,
                        deploy.yr = unique(fit.month.data$deploy.yr),
                        month = n,
                        year = unique(year(fit.month.data$date)),
                        dbbmm = list(fit.month.dbbmm),
                        ext = ext,
                        poly95 = list(fit.month.poly.95),
                        ud = list(fit.month.dbbmm.ud),
                        prj = list(fit.month.prj),
                        move = list(fit.month.move))

        # Add nth month results to storing tibble
        fit.month.hr.temp <- bind_rows(fit.month.hr.temp, df.hr)

        # Remove temporary objects from the environment to limit overwritting during loop
        rm(df.hr, fit.month.data, fit.month.move, fit.month.prj, fit.month.dbbv,
           fit.month.dbbmm, fit.month.dbbmm.sp, fit.month.dbbmm.sp.ud, fit.month.dbbmm.ud,
           fit.month.poly.95)

        # Free memory from removed objects
        gc()

        message(paste("Month", n, "done for", i))
      }

      rm(fit.month.id.data)

      gc()

      message(paste("Loop done for", i))

    }
    # Return tibble
    return(list(hr = fit.month.hr.temp))
  }


# Generate empty tibble to combine results of all deployments
fit.month.hr <- tibble(id = character(), # seal id
                       deploy.yr = double(), # deployment year
                       month = numeric(), # month in which home range occurred
                       year = numeric(), # year in which home range occurred
                       dbbmm = list(), # dynamic Brownian bridge movement model
                       ext = numeric(), # extent parameter
                       poly95 = list(), # 95% home range polygon
                       ud = list(), # utilization distribution
                       prj = list(), # projected locations
                       move = list() # move object
                       )
```

Calculate home ranges

1992

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 1992)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

1993

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 1993)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

1994

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 1994)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

1996

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 1996)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

1997

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 1997)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

1999

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 1999)
# Calculate home ranges
result <- calculate_hr(fit.yr)
```

```r
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2003

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2003)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2004

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2004)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2008

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2008)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2013

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2013)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2014

```r
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2014)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2015

```
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2015)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2017

```
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2017)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2019

```
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2019)
# Calculate home ranges
result <- calculate_hr(fit.yr)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2021

```
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2021)
# Calculate home ranges
result <- calculate_hr(fit.yr, ext.df)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

2022

```
# Select deployment
fit.yr <- filter(fit_loc_hr, deploy.yr == 2022)
# Calculate home ranges
result <- calculate_hr(fit.yr, ext.df)
# Merge with main tibble
fit.month.hr <- bind_rows(fit.month.hr, result$hr)
```

Home range characteristics

```
# fitted locations

# Extract area, shape index and coordinates of centroid from home ranges
for (i in 1:length(fit.month.hr$poly95)) {

  # Convert poly95 to sf object
  fit.month.hr$poly95[[i]] <- as(fit.month.hr$poly95[[i]], "sf")
  # Calculate area in m2
  fit.month.hr$area[i] <- st_area(fit.month.hr$poly95[[i]])
  # Calculate perimeter
  fit.month.hr$perimeter[i] <- st_perimeter(fit.month.hr$poly95[[i]])
  # Calculate shape index
  fit.month.hr$shapeindex[i] <- fit.month.hr$perimeter[i] /
    (2*sqrt(pi)*sqrt(fit.month.hr$area[i]))
  # Identify centroid
  fit.month.hr$centroid[i] <- list(st_centroid(fit.month.hr$poly95[[i]],
                                               of_largest_polygon = FALSE))
  # Set CRS to lat/long
  fit.month.hr$centroid[[i]] <- st_transform(fit.month.hr$centroid[[i]], crs = "EPSG:4326")
  # Extract longitude coordinate of centroid
  fit.month.hr$cent.x[i] <- st_coordinates(fit.month.hr$centroid[i][[1]])[1,1]
  # Extract latitude coordinate of centroid
  fit.month.hr$cent.y[i] <- st_coordinates(fit.month.hr$centroid[i][[1]])[1,2]
}

# Calculate number of locations per id/month
n.loc <- fit_loc_hr %>%
  mutate(month = month(date),
         year = year(date)) %>%
  count(year, month, id) %>%
  rename(n.loc = n)

# Calculate mean location error per id/month
loc.error <- fit_loc_hr %>%
  mutate(month = month(date),
         year = year(date)) %>%
  group_by(id, month, year) %>%
  summarise(mean.loc.error = mean(radius))

# Combine home characteristics with seals' metadata
hr.data <- fit.month.hr %>%
  # Remove unnecessary columns
  dplyr::select(-dbbmm, -poly95, -ud, -prj, -move, -perimeter) %>%
  # Join information on number of locations
  left_join(., n.loc,
            by = c("id", "year", "month")) %>%
  # Join information on mean location error
  left_join(., loc.error,
            by = c("id", "year", "month")) %>%
  # Join information on individual seals (deployment location, sex and age)
  left_join(.,
            dplyr::select(Metadata, id, location, sex, age),
            by = "id")
```

Statistical models

Prepare data for model

```
hr.data <- hr.data %>%
  mutate(
    # Convert from m2 to km2
    area = area/1e+6,
    # Convert from m to km
    mean.loc.error = mean.loc.error/1000,
    month = as.integer(month),
    # Generate period of the year variable
    period = case_when(month %in% c(1) ~ "Breeding",
                        month %in% c(2,3,4) ~ "Post_breeding",
                        month %in% c(6,7,8,9) ~ "Summer",
                        month %in% c(10,11,12) ~ "Pre_breeding"),
    period = factor(period),
    # Classify individuals as juveniles or adults
    ageclass = if_else(age >= 6, "adult", "juvenile"),
    # Generate sex-age class variable
    sexage = paste(ageclass, sex, sep = " "),
    # Generate deployment location variable based on deployment sites
    location = case_when(location == "Anticosti Island" ~ "North GSL",
                         location == "Métis-sur-Mer" ~ "North GSL",
                         location == "Brion Island" ~ "Center GSL",
                         location == "St George's Bay" ~ "South GSL",
                         location == "Amet Island" ~ "South GSL",
                         location == "Kouchibouguac" ~ "South GSL",
                         TRUE ~ location),
    )
```

Area

Model equation

$$area_{ijk} \sim Lognormal\left(\mathrm{y}_{ijk}, \sigma^2\right)$$

$$ln\left(\mathrm{y}_{ijk}\right) = \left(\beta_{m0} + ID_{m0j} + Yr_{m0k}\right) + \beta_{error_{ij}} ln\left(x_{error_{ij}}\right) + \beta_{\mathrm{period}_i} x_{\mathrm{period}_i} + \beta_{\mathrm{sexage}_j} x_{\mathrm{sexage}_j} + \beta_{\mathrm{sexage:period}_{ij}} x_{\mathrm{sexage:period}_{ij}} + f_{\mathrm{period}_i}\left(x_{yr_k}\right) + \epsilon_{ijk}$$

$$\ln\left(\sigma_{\epsilon_{ijk}}\right) = \left(\beta_{sd0_{exp}} + ID_{sd0j_{exp}} + Yr_{sd0k_{exp}}\right)$$

$$\epsilon_{ijk} \sim Normal\left(0, \sigma^2_{\epsilon_{ijk}}\right)$$

$$\begin{bmatrix} ID_{m0j} \\ Yr_{m0k} \\ ID_{sd0j_{exp}} \\ Yr_{sd0k_{exp}} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{ID_{m0}} & ... & \rho(ID_{m0j}, ID_{sd0j_{exp}})\sigma_{ID_{m0}}\sigma_{ID_{sd0_{exp}}} & ... \\ ... & \sigma^2_{Yr_{m0}} & ... & \rho(Yr_{m0j}, Yr_{sd0j_{exp}})\sigma_{Yr_{m0}}\sigma_{Yr_{sd0_{exp}}} \\ ... & ... & \sigma^2_{ID_{sd0_{exp}}} & ... \\ ... & ... & ... & \sigma^2_{Yr_{sd0}} \end{bmatrix}\right)$$

where, y_{ijk} is the ith month home range area of the jth individual in the kth year, β_{m0} is the population intercept of the mean model, ID_{m0j} is the difference between the population intercept and the random intercept for the jth individual for the mean model, Yr_{m0k} is the difference between the population intercept and the random intercept for the kth year of deployment for the mean model, $\beta_{error_{ij}}$ is the population slope of the mean error of locations used to estimate the home range ($x_{error_{ij}}$) of individual j for month j, β_{period_i} is the population slope of a categorical input variable of the period of the year (x_{period_i}) corresponding to month i for the mean model, β_{sexage_j} is the population slope of a categorical input variable of the sex-age class (x_{sexage_j}) of individual j for the mean model, $\beta_{\mathrm{sexage:period}_{ij}}$ is the population slope of the interaction between period of the year and sex-age class ($x_{\mathrm{sexage:period}_{ij}}$) of individual j in month j for the mean model, $f_{\mathrm{period}_i}\left(x_{yr_k}\right)$ is the population slope of year of deployment k (x_{yr_k}) for the period of the year corresponding to month i modelled as a thin plate regression spline for the mean model, ϵ_{ijk} is the residual error for the ith month of the jth individual in the kth year, $\sigma_{\epsilon_{ijk}}$ and $\sigma^2_{\epsilon_{ijk}}$ are the residual standard deviation and variance for the ith month of the jth individual in the kth year,

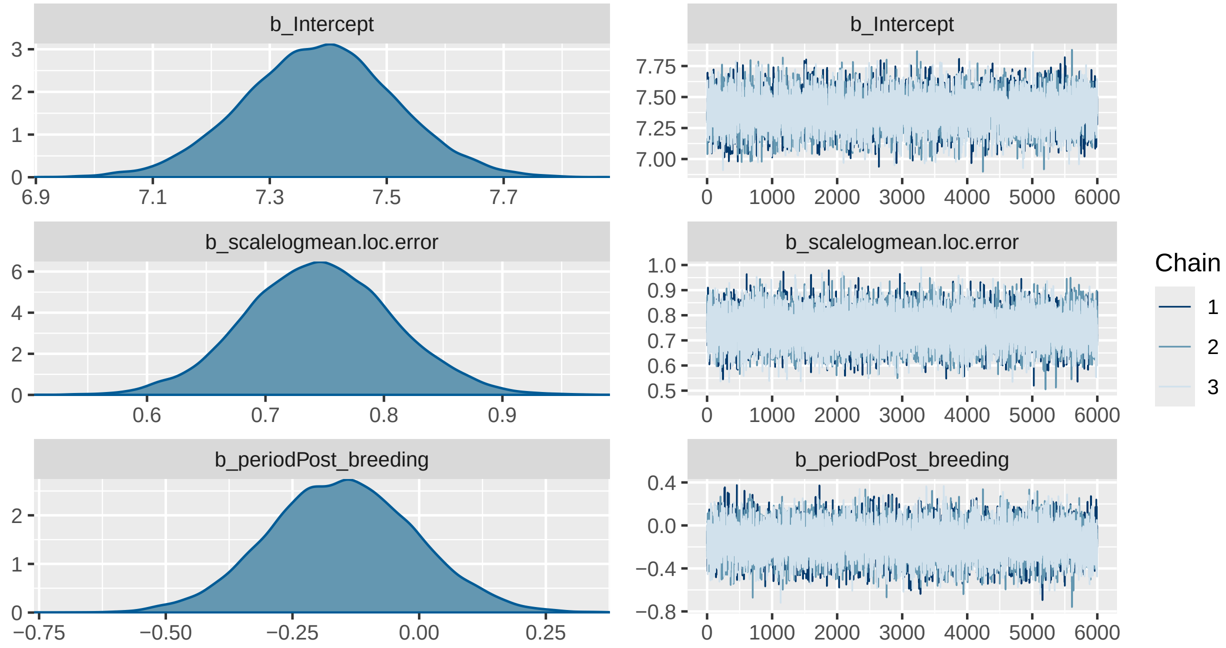
$\beta_{sd0_{exp}}$ is the population intercept of the dispersion model estimated on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $ID_{sd0j_{exp}}$ is the difference between the population intercept and the random intercept for the jth individual for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $Yr_{sd0k_{exp}}$ is the difference between the population intercept and the random intercept for the kth year of deployment for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale,

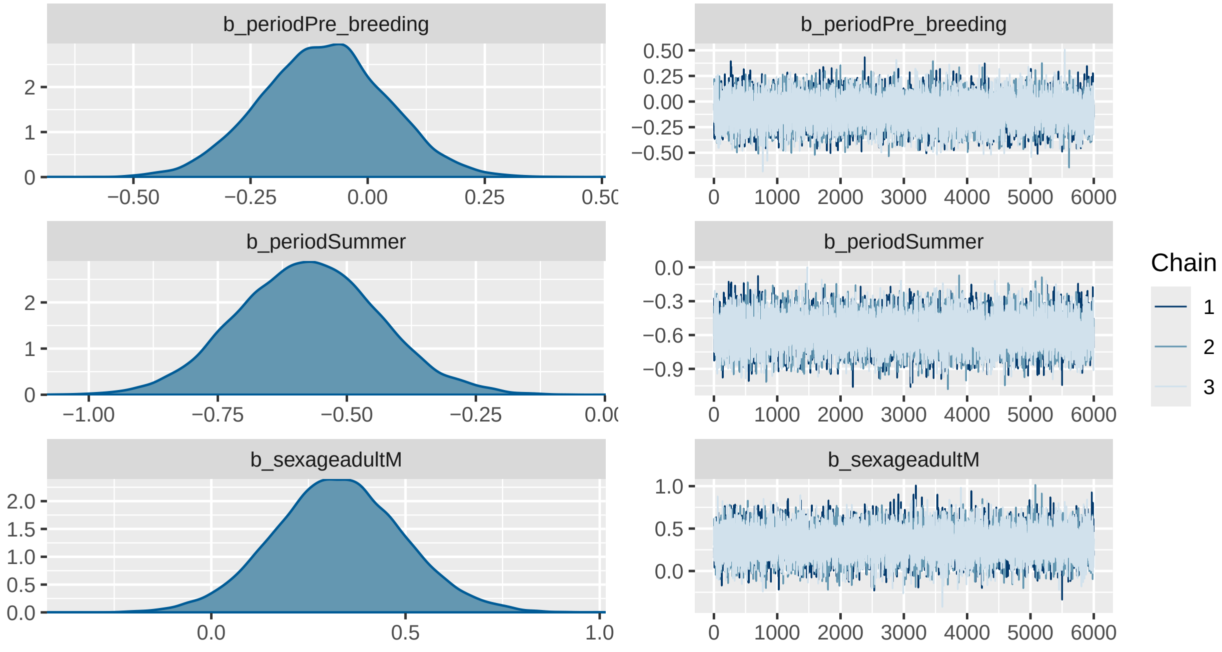
MVN is a multivariate normal distribution, $\sigma^2_{ID_{m0}}$ is the between-individual variance for the individual intercepts for the mean model, $\sigma^2_{Yr_{m0}}$ is the between-year variance for the year intercepts for the mean model, $\sigma^2_{ID_{sd0_{exp}}}$ is the between-individual variance for the individual intercepts for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $\sigma^2_{Yr_{sd0}}$ is the between-year variance for the year intercepts for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $\rho(ID_{m0j}, ID_{sd0j_{exp}})\sigma_{ID_{m0}}\sigma_{ID_{sd0_{exp}}}$ is the covariance between individual intercepts in the mean and dispersion models written as the product of the correlation between the intercepts $(\rho(ID_{m0j}, ID_{sd0j_{exp}})$ and the standard deviation for the intercepts of the mean $(\sigma_{ID_{m0}})$ and dispersion $(\sigma_{ID_{sd0_{exp}}})$ models to illustrate how the model is estimates the correlation terms, $\rho(Yr_{m0j}, Yr_{sd0j_{exp}})\sigma_{Yr_{m0}}\sigma_{Yr_{sd0_{exp}}}$ is the covariance between year intercepts in the mean and dispersion models.

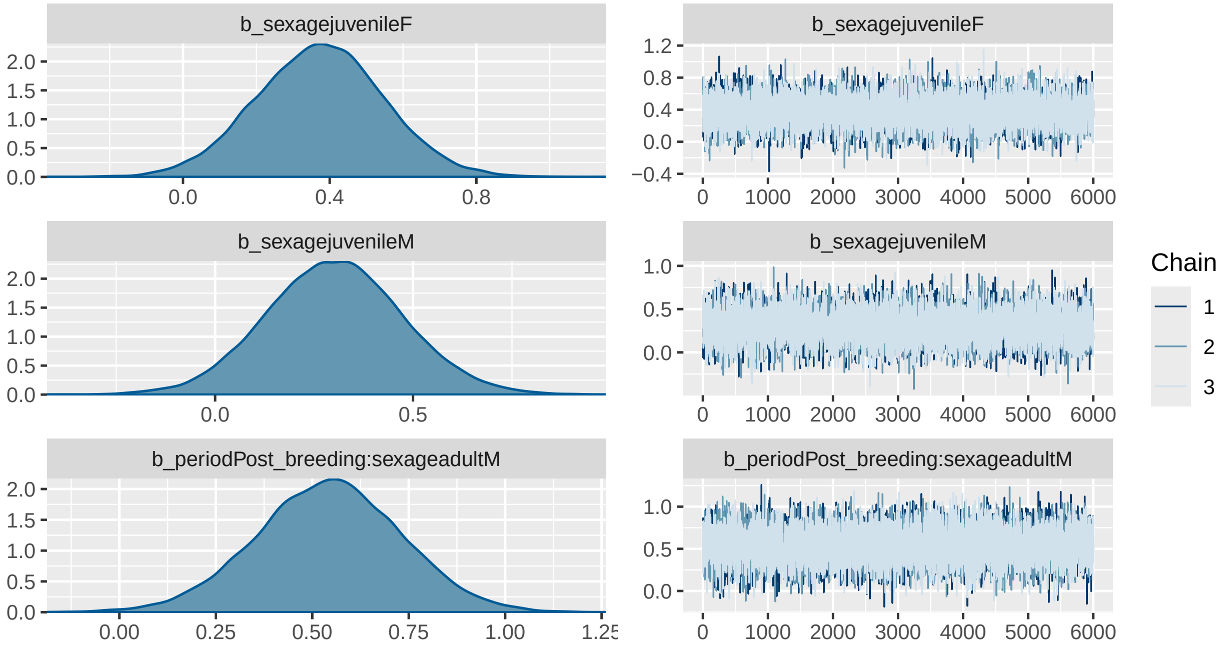
Model code

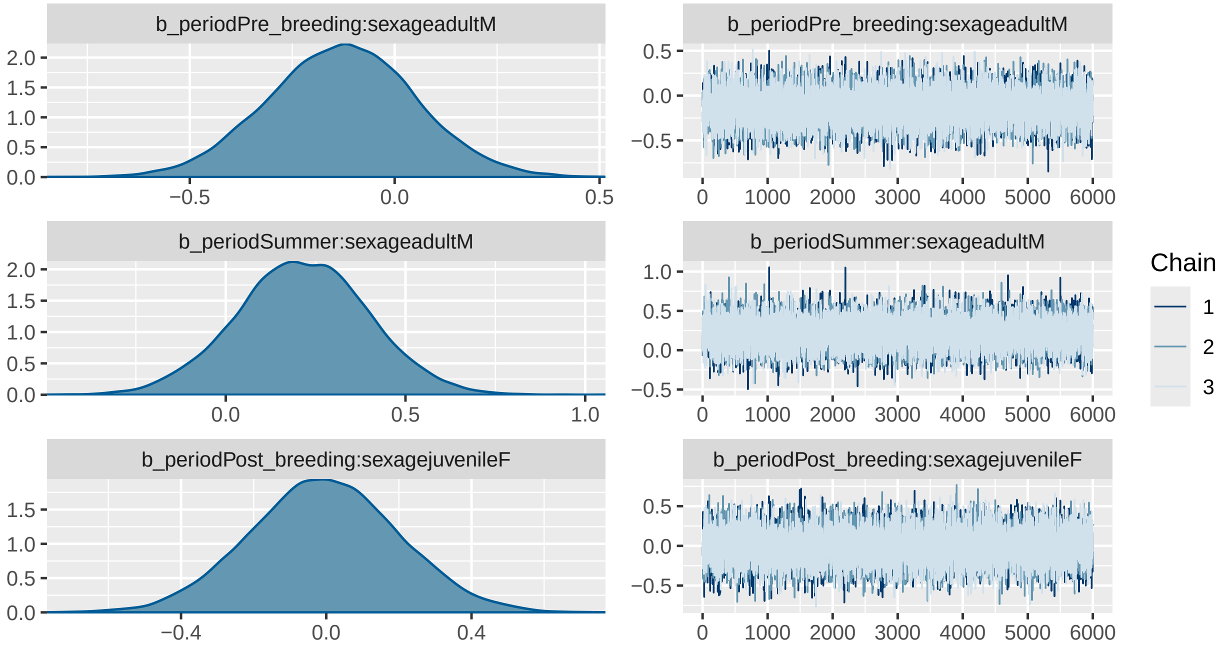
```
d_yr_area_form <- bf(
  area ~ 0 + Intercept + scale(log(mean.loc.error)) +
    period + sexage + sexage:period +
    s(scale(deploy.yr), by = period) +
    (1|p|id) + (1|y|deploy.yr),
  sigma ~ 0 + Intercept + (1|p|id) + (1|y|deploy.yr),
  family = brmsfamily("lognormal"))

d_yr_area_prior <- c(
  prior(lkj(1), class = "cor", group = "id"),

  prior(normal(7.7, 0.7), class = "b", coef = "Intercept"),

  prior(normal(0.5, 0.5), class = "b", coef = "scalelogmean.loc.error"),
  prior(normal(0, 0.7), class = "b", coef = "sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding:sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding:sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding:sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding:sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding:sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding:sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer:sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer:sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer:sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodBreeding_1"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodPost_breeding_1"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodPre_breeding_1"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodSummer_1"),

  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "id"),
  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "deploy.yr"),
  prior(normal(0, 0.7), class = "b", coef = "Intercept", dpar = "sigma"),

  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "deploy.yr", dpar = "sigma"),
  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "id", dpar = "sigma"),

  prior(normal(0, 1), class = "sds", coef = "s(scale(deploy.yr), by = period)"))

d_yr_area <- brm(
  d_yr_area_form,
  prior = d_yr_area_prior,
  data = hr.data,
  save_pars = save_pars(all = TRUE),
  sample_prior = TRUE,
  iter = 8000, cores = 3, chains = 3, warmup = 2000, thin = 1,
  control = list(adapt_delta = 0.99),
  file = "Objects/models/area.rds",
  file_refit = "on_change")
```

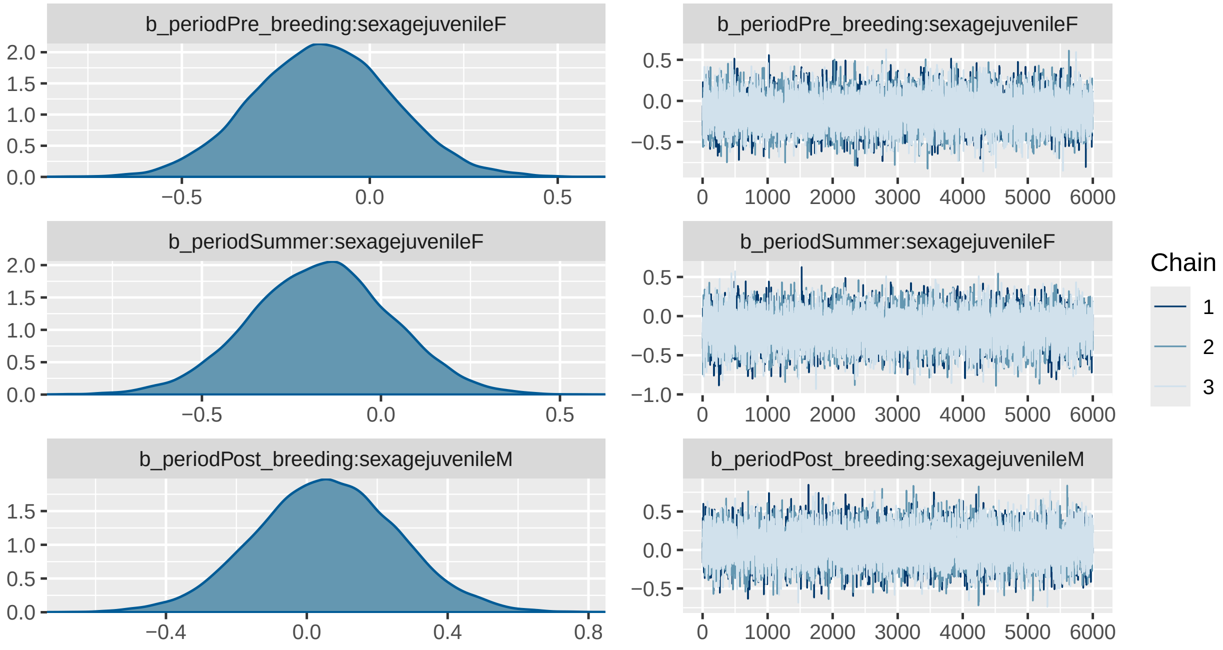
Convergence diagnostic

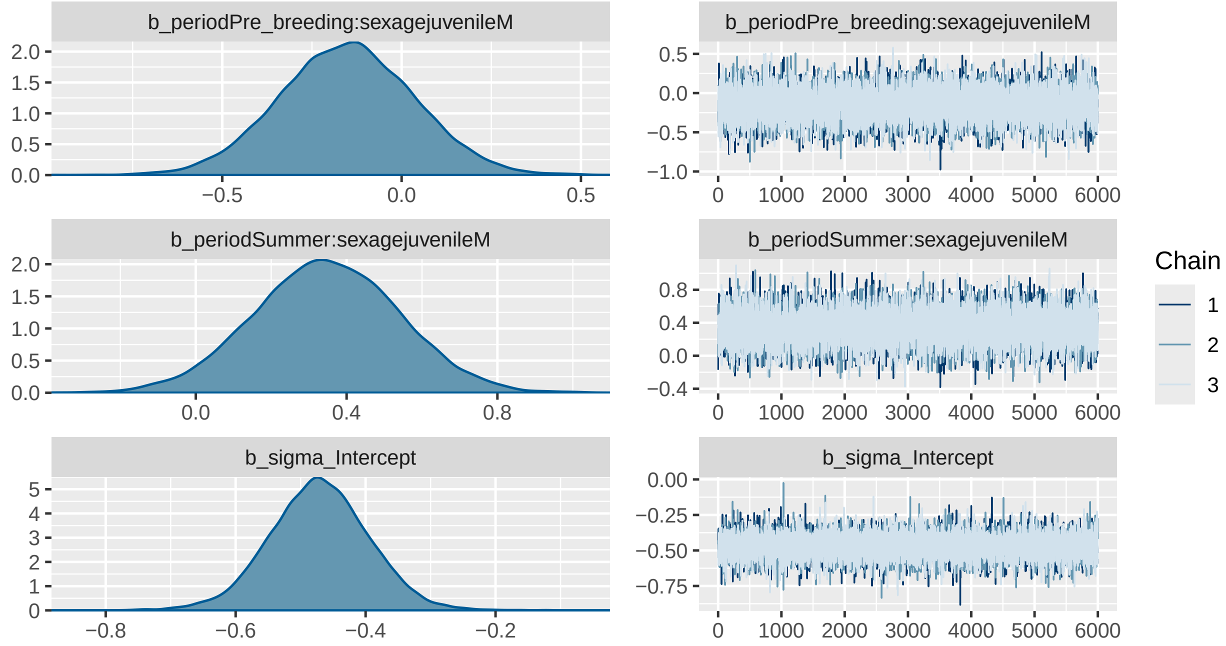


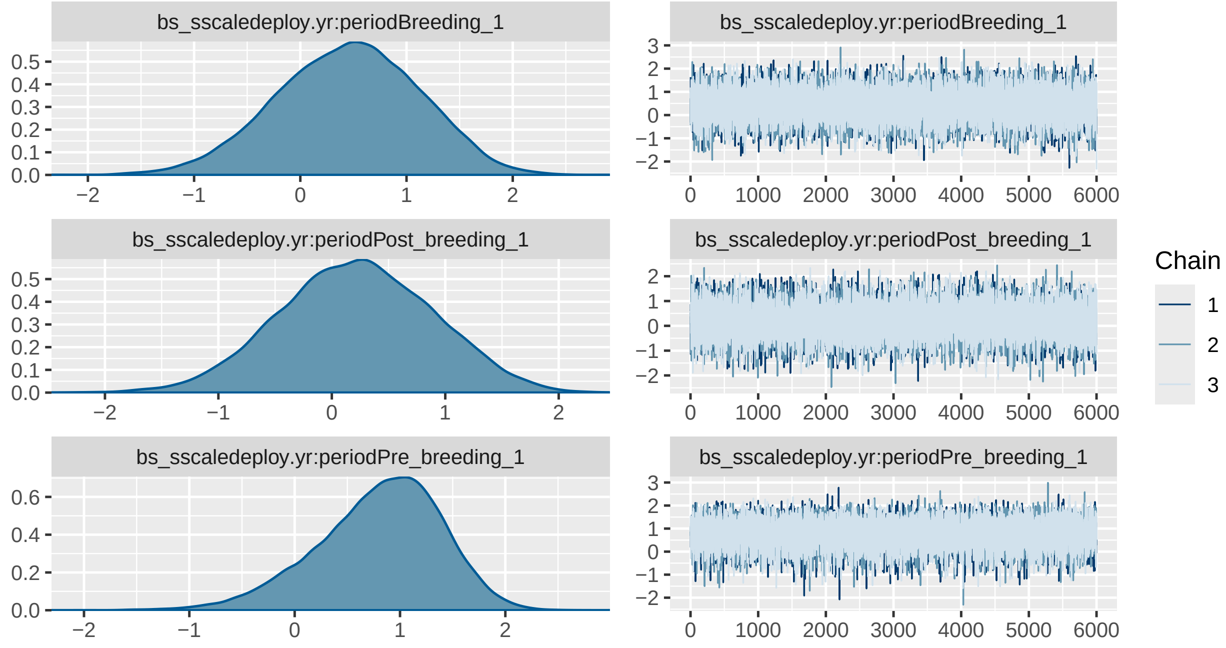

b_periodPre_breeding
2
1
0
−0.50
−0.25
0.00
0.25
0.50
b_periodPre_breeding
0.50
0.25
0.00
−0.25
−0.50
0
1000
2000
3000
4000
5000
6000
b_periodSummer
2
1
0
−1.00
−0.75
−0.50
−0.25
0.00
b_periodSummer
0.0
−0.3
−0.6
−0.9
Chain
1
2
3
b_sexageadultM
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
b_sexageadultM
1.0
0.5
0.0

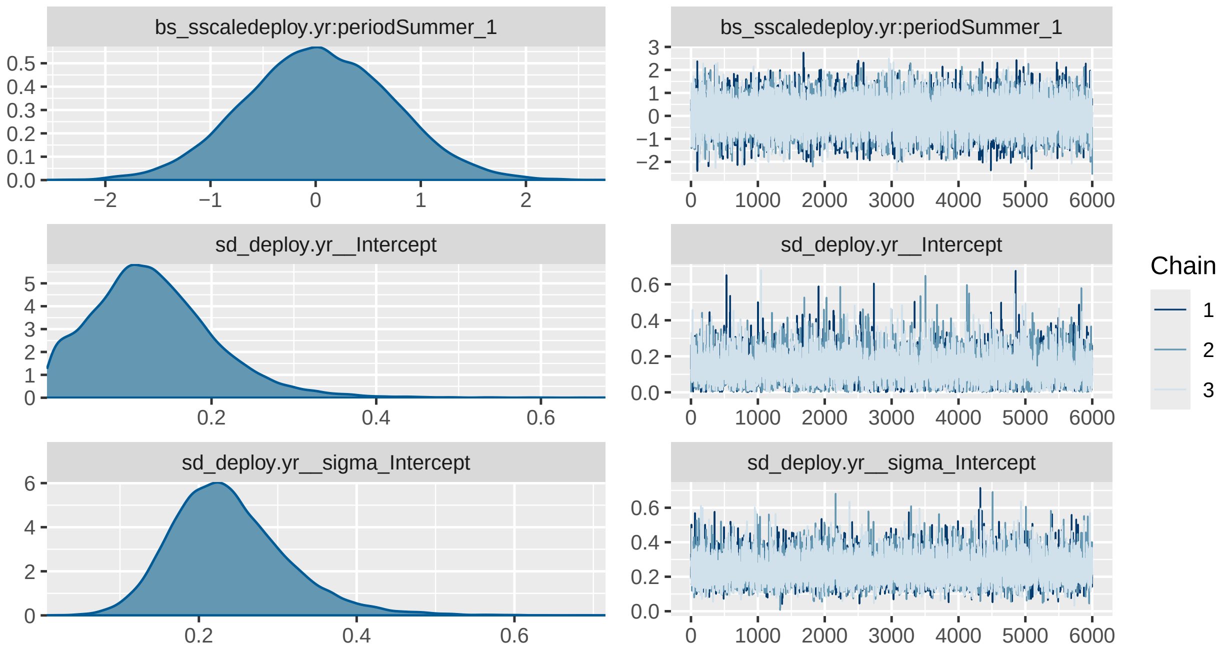

b_sexagejuvenileF
2.0
1.5
1.0
0.5
0.0
0.0
0.4
0.8
b_sexagejuvenileF
1.2
0.8
0.4
0.0
−0.4
0
1000
2000
3000
4000
5000
6000
b_sexagejuvenileM
2.0
1.5
1.0
0.5
0.0
0.0
0.5
b_sexagejuvenileM
1.0
0.5
0.0
Chain
1
2
3
b_periodPost_breeding:sexageadultM
2.0
1.5
1.0
0.5
0.0
0.00
0.25
0.50
0.75
1.00
1.25
b_periodPost_breeding:sexageadultM
1.0
0.5
0.0

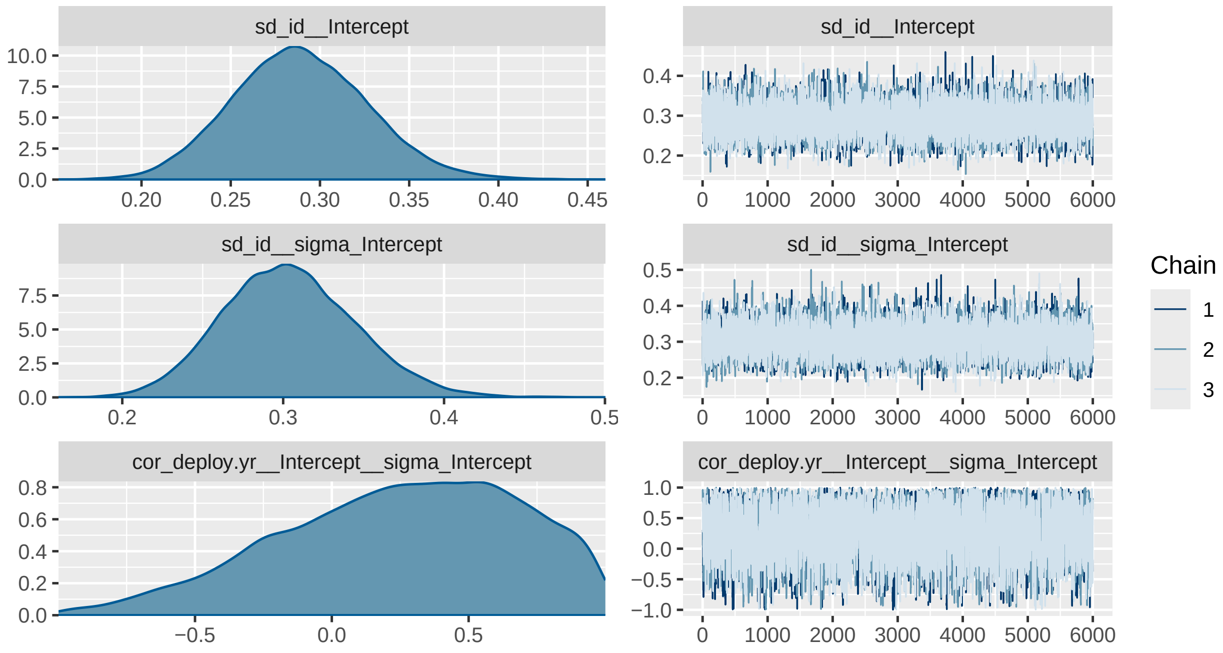

b_periodPre_breeding:sexageadultM
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
b_periodPre_breeding:sexageadultM
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
6000
b_periodSummer:sexageadultM
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
b_periodSummer:sexageadultM
1.0
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_periodPost_breeding:sexagejuvenileF
1.5
1.0
0.5
0.0
−0.4
0.0
0.4
b_periodPost_breeding:sexagejuvenileF
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
6000

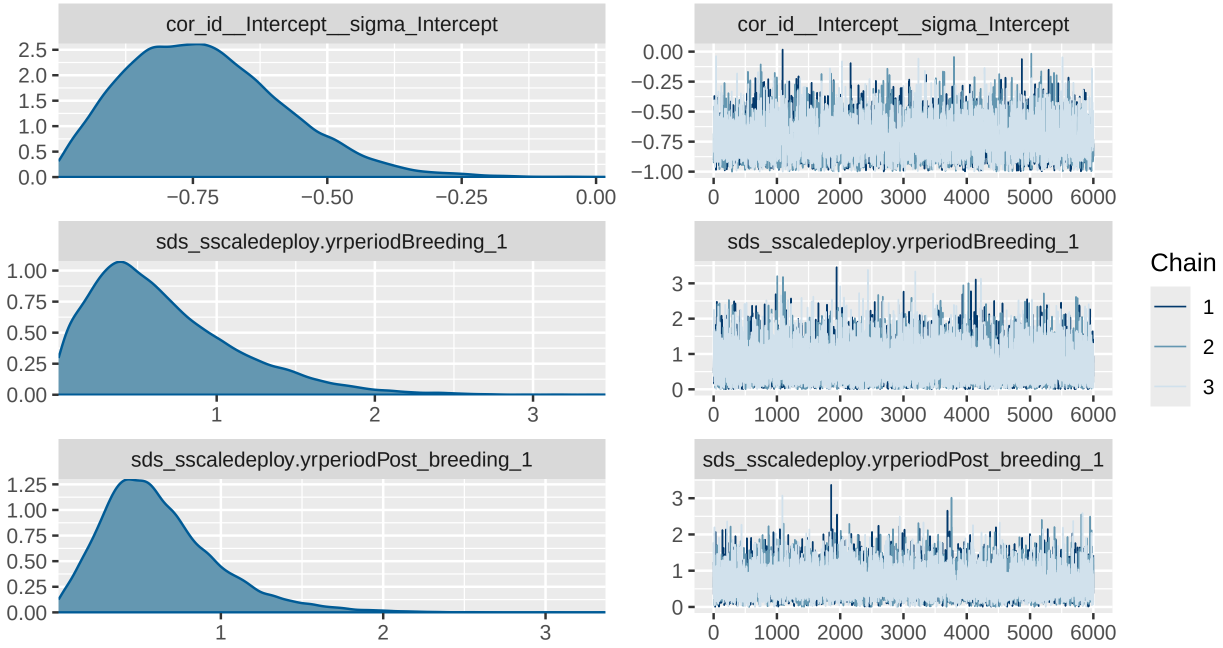

b_periodPre_breeding:sexagejuvenileF
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
b_periodPre_breeding:sexagejuvenileF
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
6000
b_periodSummer:sexagejuvenileF
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
b_periodSummer:sexagejuvenileF
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_periodPost_breeding:sexagejuvenileM
1.5
1.0
0.5
0.0
−0.4
0.0
0.4
0.8
b_periodPost_breeding:sexagejuvenileM
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
6000

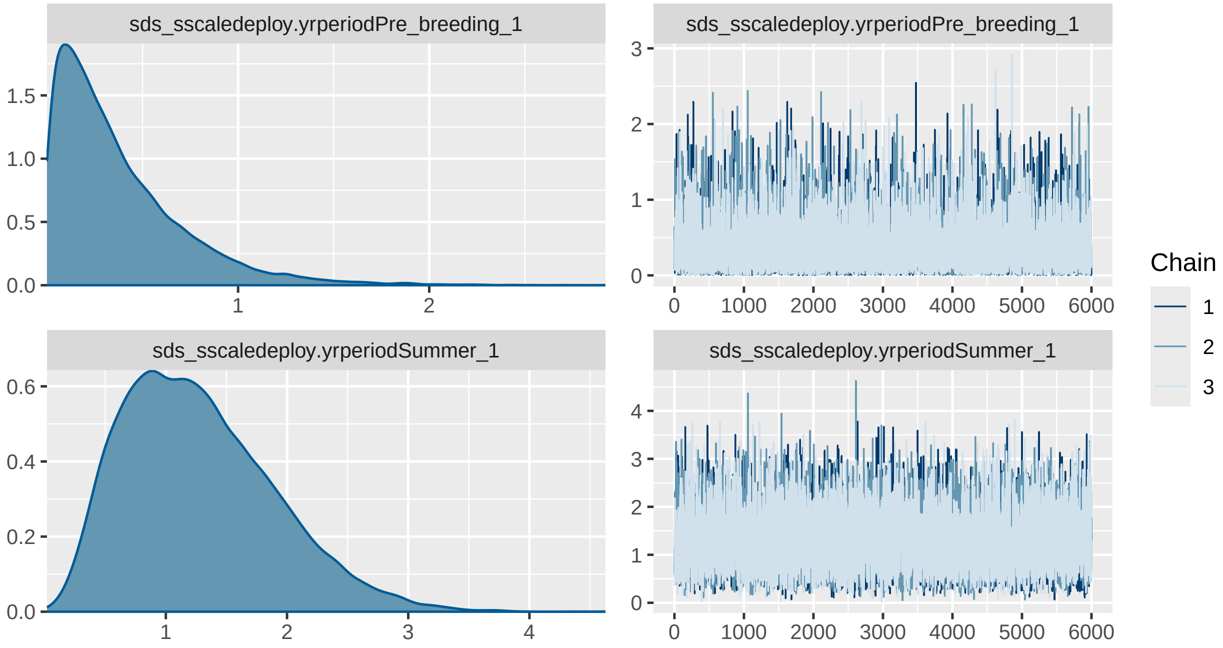

b_periodPre_breeding:sexagejuvenileM
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
b_periodPre_breeding:sexagejuvenileM
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
6000
b_periodSummer:sexagejuvenileM
2.0
1.5
1.0
0.5
0.0
0.0
0.4
0.8
b_periodSummer:sexagejuvenileM
0.8
0.4
0.0
−0.4
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_sigma_Intercept
5
4
3
2
1
0
−0.8
−0.6
−0.4
−0.2
b_sigma_Intercept
0.00
−0.25
−0.50
−0.75
0
1000
2000
3000
4000
5000
6000


bs_sscaledeploy.yr:periodBreeding_1
0.5
0.4
0.3
0.2
0.1
0.0
−2
−1
0
1
2
bs_sscaledeploy.yr:periodBreeding_1
3
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000
bs_sscaledeploy.yr:periodPost_breeding_1
0.5
0.4
0.3
0.2
0.1
0.0
−2
−1
0
1
2
bs_sscaledeploy.yr:periodPost_breeding_1
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
bs_sscaledeploy.yr:periodPre_breeding_1
0.6
0.4
0.2
0.0
−2
−1
0
1
2
bs_sscaledeploy.yr:periodPre_breeding_1
3
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000

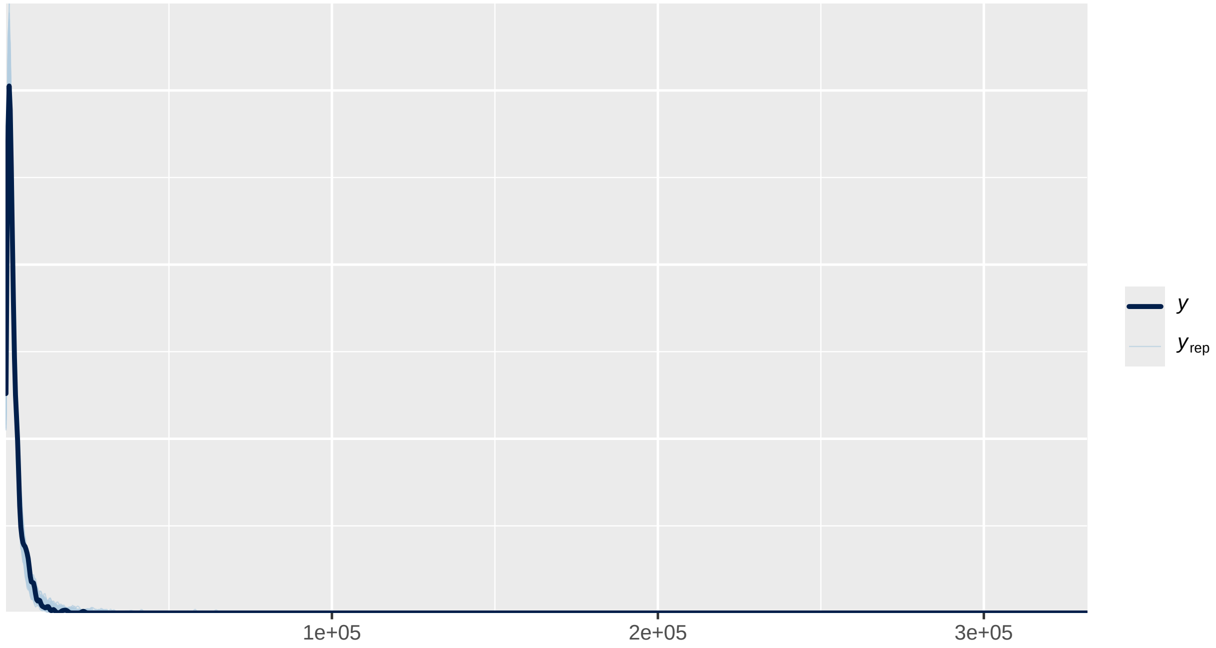

bs_sscaledeploy.yr:periodSummer_1
0.5
0.4
0.3
0.2
0.1
0.0
−2
−1
0
1
2
bs_sscaledeploy.yr:periodSummer_1
3
2
1
0
−1
−2
0 1000 2000 3000 4000 5000 6000
sd_deploy.yr__Intercept
5
4
3
2
1
0
0.2
0.4
0.6
sd_deploy.yr__Intercept
0.6
0.4
0.2
0.0
0 1000 2000 3000 4000 5000 6000
Chain
1
2
3
sd_deploy.yr__sigma_Intercept
6
4
2
0
0.2
0.4
0.6
sd_deploy.yr__sigma_Intercept
0.6
0.4
0.2
0.0
0 1000 2000 3000 4000 5000 6000


sd_id__Intercept
10.0
7.5
5.0
2.5
0.0
0.20
0.25
0.30
0.35
0.40
0.45
sd_id__Intercept
0.4
0.3
0.2
0 1000 2000 3000 4000 5000 6000
sd_id__sigma_Intercept
7.5
5.0
2.5
0.0
0.2
0.3
0.4
0.5
sd_id__sigma_Intercept
0.5
0.4
0.3
0.2
0 1000 2000 3000 4000 5000 6000
Chain
1
2
3
cor_deploy.yr__Intercept__sigma_Intercept
0.8
0.6
0.4
0.2
0.0
−0.5
0.0
0.5
cor_deploy.yr__Intercept__sigma_Intercept
1.0
0.5
0.0
−0.5
−1.0
0 1000 2000 3000 4000 5000 6000


cor_id__Intercept__sigma_Intercept
2.5
2.0
1.5
1.0
0.5
0.0
−0.75
−0.50
−0.25
0.00
cor_id__Intercept__sigma_Intercept
0.00
−0.25
−0.50
−0.75
−1.00
0
1000
2000
3000
4000
5000
6000
sds_sscaledeploy.yrperiodBreeding_1
1.00
0.75
0.50
0.25
0.00
1
2
3
sds_sscaledeploy.yrperiodBreeding_1
3
2
1
0
Chain
1
2
3
sds_sscaledeploy.yrperiodPost_breeding_1
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
sds_sscaledeploy.yrperiodPost_breeding_1
3
2
1
0
0
1000
2000
3000
4000
5000
6000

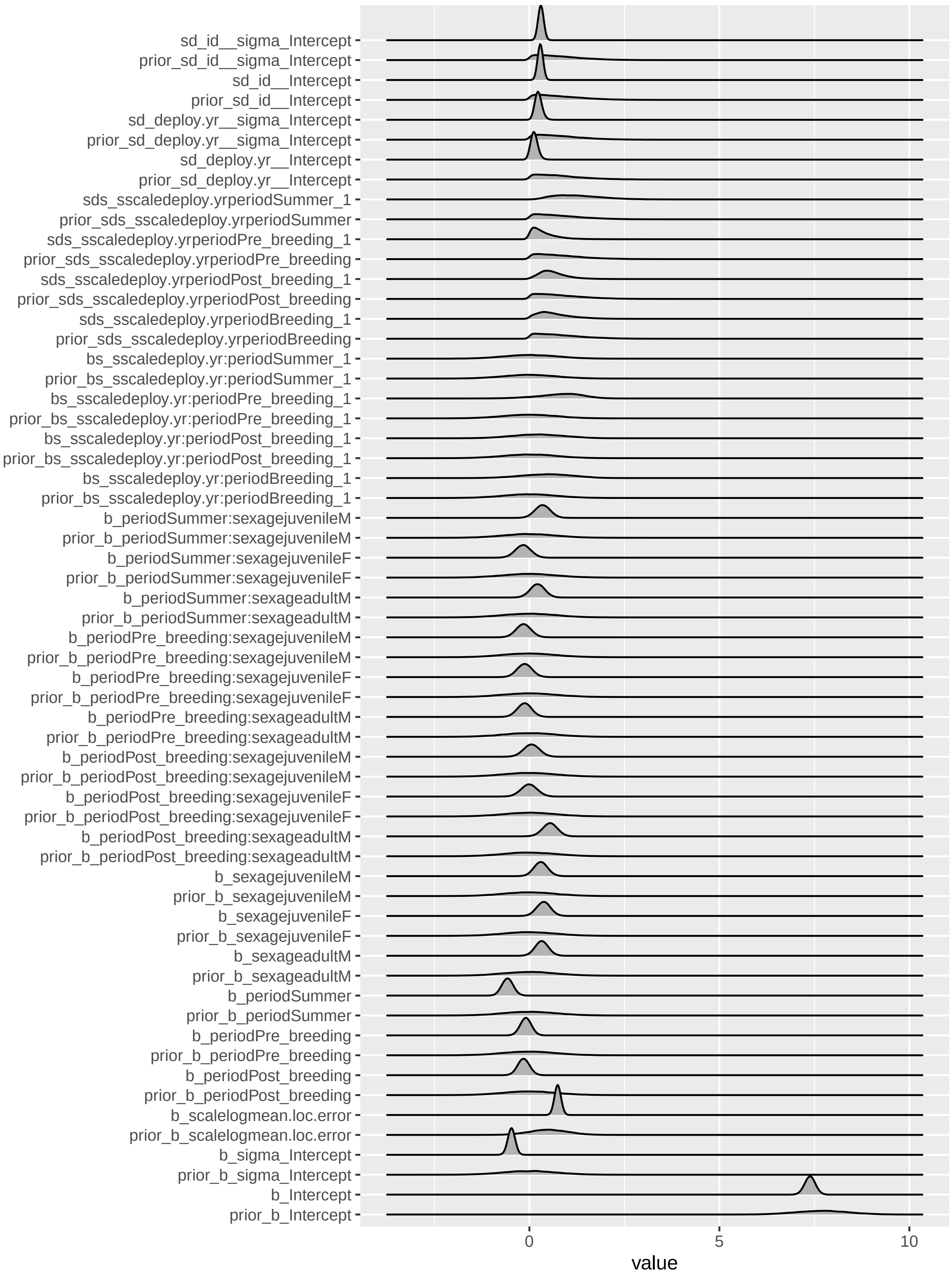

sds_sscaledeploy.yrperiodPre_breeding_1
1.5
1.0
0.5
0.0
1
2
sds_sscaledeploy.yrperiodPre_breeding_1
3
2
1
0
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
sds_sscaledeploy.yrperiodSummer_1
0.6
0.4
0.2
0.0
1
2
3
4
sds_sscaledeploy.yrperiodSummer_1
4
3
2
1
0
0
1000
2000
3000
4000
5000
6000

Posterior predictive checks



Priors vs posteriors

```
Picking joint bandwidth of 0.058
```


sd_id__sigma_Intercept
prior_sd_id__sigma_Intercept
sd_id__Intercept
prior_sd_id__Intercept
sd_deploy.yr__sigma_Intercept
prior_sd_deploy.yr__sigma_Intercept
sd_deploy.yr__Intercept
prior_sd_deploy.yr__Intercept
sds_sscaledeploy.yrperiodSummer_1
prior_sds_sscaledeploy.yrperiodSummer
sds_sscaledeploy.yrperiodPre_breeding_1
prior_sds_sscaledeploy.yrperiodPre_breeding
sds_sscaledeploy.yrperiodPost_breeding_1
prior_sds_sscaledeploy.yrperiodPost_breeding
sds_sscaledeploy.yrperiodBreeding_1
prior_sds_sscaledeploy.yrperiodBreeding
bs_sscaledeploy.yr:periodSummer_1
prior_bs_sscaledeploy.yr:periodSummer_1
bs_sscaledeploy.yr:periodPre_breeding_1
prior_bs_sscaledeploy.yr:periodPre_breeding_1
bs_sscaledeploy.yr:periodPost_breeding_1
prior_bs_sscaledeploy.yr:periodPost_breeding_1
bs_sscaledeploy.yr:periodBreeding_1
prior_bs_sscaledeploy.yr:periodBreeding_1
b_periodSummer:sexagejuvenileM
prior_b_periodSummer:sexagejuvenileM
b_periodSummer:sexagejuvenileF
prior_b_periodSummer:sexagejuvenileF
b_periodSummer:sexageadultM
prior_b_periodSummer:sexageadultM
b_periodPre_breeding:sexagejuvenileM
prior_b_periodPre_breeding:sexagejuvenileM
b_periodPre_breeding:sexagejuvenileF
prior_b_periodPre_breeding:sexagejuvenileF
b_periodPre_breeding:sexageadultM
prior_b_periodPre_breeding:sexageadultM
b_periodPost_breeding:sexagejuvenileM
prior_b_periodPost_breeding:sexagejuvenileM
b_periodPost_breeding:sexagejuvenileF
prior_b_periodPost_breeding:sexagejuvenileF
b_periodPost_breeding:sexageadultM
prior_b_periodPost_breeding:sexageadultM
b_sexagejuvenileM
prior_b_sexagejuvenileM
b_sexagejuvenileF
prior_b_sexagejuvenileF
b_sexageadultM
prior_b_sexageadultM
b_periodSummer
prior_b_periodSummer
b_periodPre_breeding
prior_b_periodPre_breeding
b_periodPost_breeding
prior_b_periodPost_breeding
b_scalelogmean.loc.error
prior_b_scalelogmean.loc.error
b_sigma_Intercept
prior_b_sigma_Intercept
b_Intercept
prior_b_Intercept
0
5
10
value

Model summary

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Population-Level Effects:							
Intercept	7.385	0.129	7.131	7.639	1.000	5 889	9 433
scalelogmean.loc.error	0.746	0.062	0.625	0.869	1.000	12 782	13 452
periodPost_breeding	-0.152	0.144	-0.433	0.134	1.000	7 155	10 882
periodPre_breeding	-0.094	0.135	-0.357	0.169	1.000	6 218	10 210
periodSummer	-0.575	0.137	-0.842	-0.298	1.000	6 406	9 872
sexageadultM	0.327	0.167	0.001	0.660	1.000	6 286	9 959
sexagejuvenileF	0.374	0.174	0.032	0.716	1.000	7 126	11 083
sexagejuvenileM	0.302	0.173	-0.036	0.646	1.000	6 496	10 532
periodPost_breeding:sexageadultM	0.542	0.189	0.167	0.910	1.000	8 181	11 880
periodPre_breeding:sexageadultM	-0.129	0.180	-0.483	0.223	1.000	7 443	11 420
periodSummer:sexageadultM	0.214	0.184	-0.150	0.572	1.000	7 471	10 362
periodPost_breeding:sexagejuvenileF	-0.006	0.204	-0.403	0.395	1.000	9 080	11 997
periodPre_breeding:sexagejuvenileF	-0.125	0.186	-0.486	0.239	1.000	7 721	11 091
periodSummer:sexagejuvenileF	-0.167	0.197	-0.555	0.215	1.000	7 970	11 512
periodPost_breeding:sexagejuvenileM	0.057	0.202	-0.340	0.457	1.000	8 428	11 582
periodPre_breeding:sexagejuvenileM	-0.153	0.188	-0.521	0.216	1.000	7 096	11 439
periodSummer:sexagejuvenileM	0.347	0.193	-0.033	0.725	1.000	7 379	11 203
sigma_Intercept	-0.472	0.077	-0.625	-0.322	1.000	11 296	12 397
sscaledeploy.yr:periodBreeding_1	0.447	0.671	-0.895	1.687	1.000	14 247	12 897
sscaledeploy.yr:periodPost_breeding_1	0.217	0.676	-1.101	1.511	1.000	15 784	10 083
sscaledeploy.yr:periodPre_breeding_1	0.799	0.592	-0.495	1.802	1.000	12 184	13 922
sscaledeploy.yr:periodSummer_1	0.032	0.699	-1.325	1.411	1.001	22 500	11 573
Smooth Terms:							
sds(sscaledeploy.yrperiodBreeding_1)	0.666	0.463	0.044	1.794	1.000	6 069	7 442
sds(sscaledeploy.yrperiodPost_breeding_1)	0.625	0.360	0.097	1.513	1.000	8 852	7 750
sds(sscaledeploy.yrperiodPre_breeding_1)	0.374	0.334	0.013	1.254	1.000	6 818	9 314
sds(sscaledeploy.yrperiodSummer_1)	1.275	0.619	0.330	2.661	1.000	7 204	8 061
Group-Level Effects:							
~Id (Number of levels: 124)							
sd(Intercept_id)	0.289	0.038	0.217	0.366	1.000	7 565	11 414
sd(sigma_Intercept_id)	0.304	0.041	0.228	0.389	1.000	7 182	11 185
cor(Intercept_id,sigma_Intercept_id)	-0.726	0.148	-0.964	-0.401	1.001	3 531	4 907
~Year (Number of levels: 16)							
sd(Intercept_yr)	0.130	0.075	0.010	0.300	1.001	3 538	5 352
sd(sigma_Intercept_yr)	0.239	0.074	0.116	0.409	1.000	6 608	8 498
cor(Intercept_yr,sigma_Intercept_yr)	0.250	0.432	-0.664	0.939	1.001	3 104	3 868

Post-hoc tests

```
# 'CI': 95%-CI
# '*': The value tested against lies outside the 95%-CI.
# Evidence ratio is a Bayes factor between the hypothesis and its alternative.
#   Values greater than one indicate that evidence in favor of the point
#   hypothesis has increased after seeing the data.

####################################################################################################################
# Post-hoc comparison between periods, within sexage classes #######################################################
####################################################################################################################
## Adult females
# Define hypotheses
v_af <- c("Summer = Breeding" = "periodSummer = 0",
          "Summer =  Pre-breeding" = "periodSummer = periodPre_breeding",
          "Summer = Post-breeding" = "periodSummer = periodPost_breeding",
          "Pre-breeding = Breeding" = "periodPre_breeding = 0",
          "Pre-breeding = Post-breeding" = "periodPre_breeding = periodPost_breeding",
          "Breeding = Post-breeding" = "0 = periodPost_breeding")
# Test hypotheses
h_af <- hypothesis(d_yr_area, v_af)
# Store results
h_af_tib <- h_af$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Adult males
# Define hypotheses
v_am <- c("Summer = Breeding" =
            "periodSummer + periodSummer:sexageadultM = 0",
          "Summer =  Pre-breeding" =
            "periodSummer + periodSummer:sexageadultM = periodPre_breeding + periodPre_breeding:sexageadultM",
          "Summer = Post-breeding" =
            "periodSummer + periodSummer:sexageadultM = periodPost_breeding + periodPost_breeding:sexageadultM",
          "Pre-breeding = Breeding" =
            "periodPre_breeding + periodPre_breeding:sexageadultM = 0",
          "Pre-breeding = Post-breeding" =
            "periodPre_breeding + periodPre_breeding:sexageadultM = periodPost_breeding +
             periodPost_breeding:sexageadultM",
          "Breeding = Post-breeding" =
            "0 = periodPost_breeding + periodPost_breeding:sexageadultM")
# Test hypotheses
h_am <- hypothesis(d_yr_area, v_am)
# Store results
h_am_tib <- h_am$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile females
# Define hypotheses
v_jf <- c("Summer = Breeding" =
            "periodSummer + periodSummer:sexagejuvenileF = 0",
          "Summer =  Pre-breeding" =
            "periodSummer + periodSummer:sexagejuvenileF = periodPre_breeding + periodPre_breeding:sexagejuvenileF",
          "Summer = Post-breeding" =
            "periodSummer + periodSummer:sexagejuvenileF = periodPost_breeding + periodPost_breeding:sexagejuvenileF",
          "Pre-breeding = Breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileF = 0",
          "Pre-breeding = Post-breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileF = periodPost_breeding +
             periodPost_breeding:sexagejuvenileF",
          "Breeding = Post-breeding" =
            "0 = periodPost_breeding + periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_jf <- hypothesis(d_yr_area, v_jf)
# Store results
h_jf_tib <- h_jf$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
```

```r
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile males
# Define hypotheses
v_jm <- c("Summer = Breeding" =
            "periodSummer + periodSummer:sexagejuvenileM = 0",
          "Summer =  Pre-breeding" =
            "periodSummer + periodSummer:sexagejuvenileM = periodPre_breeding + periodPre_breeding:sexagejuvenileM",
          "Summer = Post-breeding" =
            "periodSummer + periodSummer:sexagejuvenileM = periodPost_breeding + periodPost_breeding:sexagejuvenileM",
          "Pre-breeding = Breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileM = 0",
          "Pre-breeding = Post-breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileM = periodPost_breeding +
             periodPost_breeding:sexagejuvenileM",
          "Breeding = Post-breeding" =
            "0 = periodPost_breeding + periodPost_breeding:sexagejuvenileM")
# Test hypotheses
h_jm <- hypothesis(d_yr_area, v_jm)
# Store results
h_jm_tib <- h_jm$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_area_seas <-
  full_join(h_af_tib, h_am_tib, by = "Hypothesis") %>%
  full_join(., h_jf_tib, by = "Hypothesis") %>%
  full_join(., h_jm_tib, by = "Hypothesis")

####################################################################################################################
# Post-hoc comparison between sexage classes, within periods ######################################################
####################################################################################################################
## Summer
# Define hypotheses
v_sum <- c("Adult F = Adult M" =
             "0 = sexageadultM + periodSummer:sexageadultM",
          "Adult F = Juvenile F" =
            "0 = sexagejuvenileF + periodSummer:sexagejuvenileF",
          "Adult F = Juvenile M" =
            "0 = sexagejuvenileM + periodSummer:sexagejuvenileM",
          "Adult M = Juvenile M" =
            "sexageadultM + periodSummer:sexageadultM = sexagejuvenileM + periodSummer:sexagejuvenileM",
          "Adult M = Juvenile F" =
            "sexageadultM + periodSummer:sexageadultM = sexagejuvenileF + periodSummer:sexagejuvenileF",
          "Juvenile M = Juvenile F" =
            "sexagejuvenileM + periodSummer:sexagejuvenileM = sexagejuvenileF + periodSummer:sexagejuvenileF")
# Test hypotheses
h_sum <- hypothesis(d_yr_area, v_sum)
# Store results
h_sum_tib <- h_sum$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Pre-breeding
# Define hypotheses
v_pre <- c("Adult F = Adult M" =
             "0 = sexageadultM + periodPre_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = sexagejuvenileF + periodPre_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = sexagejuvenileM + periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "sexageadultM + periodPre_breeding:sexageadultM = sexagejuvenileM + periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "sexageadultM + periodPre_breeding:sexageadultM = sexagejuvenileF + periodPre_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "sexagejuvenileM + periodPre_breeding:sexagejuvenileM = sexagejuvenileF +
              periodPre_breeding:sexagejuvenileF")
# Test hypotheses
h_pre <- hypothesis(d_yr_area, v_pre)
# Store results
h_pre_tib <- h_pre$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Breeding
# Define hypotheses
v_bred <- c("Adult F = Adult M" = "0 = sexageadultM ",
           "Adult F = Juvenile F" = "0 = sexagejuvenileF",
           "Adult F = Juvenile M" = "0 = sexagejuvenileM",
           "Adult M = Juvenile M" = "sexageadultM = sexagejuvenileM",
           "Adult M = Juvenile F" = "sexageadultM = sexagejuvenileF",
           "Juvenile M = Juvenile F" = "sexagejuvenileM = sexagejuvenileF")
# Test hypotheses
h_bred <- hypothesis(d_yr_area, v_bred)
# Store results
h_bred_tib <- h_bred$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Post-breeding
# Define hypotheses
v_post <- c("Adult F = Adult M" =
              "0 = sexageadultM + periodPost_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = sexagejuvenileF + periodPost_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = sexagejuvenileM + periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "sexageadultM + periodPost_breeding:sexageadultM = sexagejuvenileM + periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "sexageadultM + periodPost_breeding:sexageadultM = sexagejuvenileF + periodPost_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "sexagejuvenileM + periodPost_breeding:sexagejuvenileM = sexagejuvenileF +
              periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_post <- hypothesis(d_yr_area, v_post)
# Store results
h_post_tib <- h_post$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_area_onto <-
  full_join(h_sum_tib, h_pre_tib, by = "Hypothesis") %>%
  full_join(., h_bred_tib, by = "Hypothesis") %>%
  full_join(., h_post_tib, by = "Hypothesis")
```

Individual differences

Equations

Agreement repeatabilities, adjusted repeatabilities and coefficients of variation were calculated based on the tutorial provided by O'dea et al. 2022.

In a DHGLM, residual variation can be modelled using residual standard deviation or residual variance as the response variable of the dispersion model. brms uses residual standard deviation as the response variable for the dispersion model ($\ln\left(\sigma_{\epsilon_{ijk}}\right)$). We converted the estimates of the dispersion models from the standard deviation scale to the variance scale ($\ln\left(\sigma^2_{\epsilon_{ijk}}\right)$) before calculating repeatabilities and coefficients of variation. The relation between the scales is $\ln\left(\sigma_{\epsilon_{ijk}}\right) = \frac{1}{2}\ln\left(\sigma^2_{\epsilon_{ijk}}\right)$. We thus converted the estimates of the dispersion model (standard deviation scale) to variance scale by multiplying them by 2:

$$\begin{aligned}
\beta_{v0_{exp}} &= 2\beta_{sd0_{exp}}, \\
\sigma_{ID_{v0_{exp}}} &= 2\sigma_{ID_{sd0_{exp}}}, \\
\sigma_{Yr_{v0_{exp}}} &= 2\sigma_{Yr_{sd0_{exp}}},
\end{aligned}$$

where, $\beta_{sd0_{exp}}$ and $\beta_{v0_{exp}}$ are the population intercepts of the dispersion model on the standard deviation and variance scales, $\sigma_{ID_{sd0_{exp}}}$ and $\sigma_{ID_{v0_{exp}}}$ are the standard deviations for the individual intercepts of the dispersion model on the standard deviation and variance scales, $\sigma_{Yr_{sd0_{exp}}}$ and $\sigma_{Yr_{v0_{exp}}}$ are the standard deviations for the year intercepts of the dispersion model on the standard deviation and variance scales.

Agreement repeatability for mean model (Rp_m):

$$Rp_m = \frac{\sigma^2_{ID_m}}{\sigma^2_p},$$

$$\begin{aligned}
\sigma^2_p &= \sigma^2_{ID_m} + \sigma^2_{Yr_m} + \sigma^2_{\beta_m} + \bar{\sigma}^2_\varepsilon, \\
\bar{\sigma}^2_\varepsilon &= e^{\left(\beta_{v0_{exp}} + \frac{\sigma^2_{ID_{v0_{exp}}} + \sigma^2_{Yr_{v0_{exp}}}}{2}\right)},
\end{aligned}$$

where, $\sigma^2_{ID_m}$ is the variance component for the between-individual differences in the mean model, σ^2_p is the total phenotypic variance, $\sigma^2_{\beta_m}$ is the variance component for the population-level effects in the mean model, $\sigma^2_{Yr_m}$ is the variance component for the between year differences in the mean model, $\sigma^2_{\beta_m}$ is the variance component for the population-level effects in the mean model, $\bar{\sigma}^2_\varepsilon$ is the average residual variance, $\beta_{v0_{exp}}$ is the population intercept of the dispersion model on the ln(var) scale, $\sigma^2_{ID_{v0_{exp}}}$ is the is the between-individual variance for the individual intercepts for the dispersion model on the ln(var) scale, $\sigma^2_{Yr_{v0_{exp}}}$ is the between-year variance for the year intercepts for the dispersion model on the ln(var) scale.

Adjusted repeatability for mean model ($Rp\ adj_m$):

$$Rp\ adj_m = \frac{\sigma^2_{ID_m}}{\sigma^2_{p_{adj}}},$$

$$\sigma^2_{p_{adj}} \;=\; \sigma^2_{ID_m} + \sigma^2_{Yr_m} + \bar{\sigma}^2_{\varepsilon},$$

where, $\sigma^2_{ID_m}$ is the variance component for the between-individual differences in the mean model, $\sigma^2_{p_{adj}}$ is the adjusted phenotypic variance (i.e. phenotypic variance not explained by population-level effects), $\sigma^2_{Yr_m}$ is the variance component for the between year differences in the mean model, $\sigma^2_{\beta_m}$ is the variance component for the population-level effects in the mean model, $\bar{\sigma}^2_{\varepsilon}$ is the average residual variance.

Coefficient of variation for the mean model (CV_m):

$$CV_m = \frac{\sigma_{ID_m}}{\mu_P}$$

where, σ_{ID_m} is the standard deviation for between-individual differences in the mean model, μ_P is the average individual phenotype.

Agreement repeatability for the dispersion model (Rp_v):

$$Rp_v = \frac{\sigma^2_{ID_v}}{\sigma^2_{\sigma^2_p}},$$

$$\sigma^2_{ID_v} \;=\; \sigma^2_{\sigma^2_p}\left(\frac{\sigma^2_{ID_{v0_{exp}}}}{\sigma^2_{ID_{v0_{exp}}} \;+\; \sigma^2_{Yr_{v0_{exp}}}}\right),$$

$$\sigma^2_{\sigma^2_p} \;=\; 2\left(\sigma^2_p\right)^2 \;+\; 3\sigma^2_{\sigma^2_\varepsilon},$$

$$\sigma^2_{\sigma^2_\varepsilon} \;=\; \left(e^{\left(\sigma^2_{ID_{v0_{exp}}} \;+\; \sigma^2_{Yr_{v0_{exp}}}\right)} \;-\; 1\right) e^{\left(2\beta_{v0_{exp}} \;+\; \sigma^2_{ID_{v0_{exp}}} \;+\; \sigma^2_{Yr_{v0_{exp}}}\right)},$$

where, $\sigma^2_{ID_v}$ is the between individual variance in residual variation on the observed, i.e. mean model, scale, $\sigma^2_{\sigma^2_p}$ is the total variance in phenotypic variance on the observed scale, $\sigma^2_{ID_{v0_{exp}}}$ is the between-individual variance for the individual intercepts of the dispersion model on the ln(var) scale, $\sigma^2_{Yr_{v0_{exp}}}$ is the between-year variance for the year intercepts for the dispersion model on the ln(var) scale, σ^2_p is the total phenotypic variance, $\sigma^2_{\sigma^2_\varepsilon}$ is the variance in residual variation on the observed scale, , $\beta_{v0_{exp}}$ is the population intercept of the dispersion model on the ln(var) scale, $\sigma^2_{ID_{v0_{exp}}}$ is the between-individual variance for the individual intercepts for the dispersion model on the ln(var) scale, $\sigma^2_{Yr_{v0_{exp}}}$ is the between-year variance for the year intercepts for the dispersion model on the ln(var) scale.

Coefficient of variation for the dispersion model (CV_v):

$$CV_v = \frac{\sigma_{ID_v}}{\bar{\sigma}^2_{\varepsilon}},$$

where, σ_{ID_v} is the standard deviation for between-individual differences for the dispersion model on the ln(var) scale, $\bar{\sigma}^2_{\varepsilon}$ is the average residual variance.

Code

```
####################################################################################################################
#### Define functions ##############################################################################################
####################################################################################################################
# Function for summing variance components
func_sum_var <- function(vars) {
  sum_var <- sapply(1:nrow(vars), function(x) sum(vars[x, ], na.rm = T))
  return(sum_var)
}

####################################################################################################################
#### Extract parameters from Mean and Dispersion models ############################################################
####################################################################################################################
## Extract the population coefficients for mean model
# Adult females
Baf_data <- tibble(mean.loc.error = mean(hr.data$mean.loc.error),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "adult F",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Baf_all <- posterior_linpred(d_yr_area, newdata = Baf_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Baf <- Baf_all[,1]*(4/11) + Baf_all[,2]*(3/11) + Baf_all[,3]*(1/11) +
  Baf_all[,4]*(3/11)

# Adult males
Bam_data <- tibble(mean.loc.error = mean(hr.data$mean.loc.error),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "adult M",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Bam_all <- posterior_linpred(d_yr_area, newdata = Bam_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bam <- Bam_all[,1]*(4/11) + Bam_all[,2]*(3/11) + Bam_all[,3]*(1/11) +
  Bam_all[,4]*(3/11)

# Juvenile females
Bjf_data <- tibble(mean.loc.error = mean(hr.data$mean.loc.error),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "juvenile F",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Bjf_all <- posterior_linpred(d_yr_area, newdata = Bjf_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bjf <- Bjf_all[,1]*(4/11) + Bjf_all[,2]*(3/11) + Bjf_all[,3]*(1/11) +
  Bjf_all[,4]*(3/11)

# Juvenile males
Bjm_data <- tibble(mean.loc.error = mean(hr.data$mean.loc.error),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "juvenile M",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Bjm_all <- posterior_linpred(d_yr_area, newdata = Bjm_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bjm <- Bjm_all[,1]*(4/11) + Bjm_all[,2]*(3/11) + Bjm_all[,3]*(1/11) +
  Bjm_all[,4]*(3/11)

## Calculate the population intercept (average individual phenotype)
## for the mean model (B_pm) and the dispersion model (B_pv)
# Mean model (assuming equal representation of each sexage class)
B_pm <- Baf*(1/4) + Bam*(1/4) + Bjf*(1/4) + Bjm*(1/4)
# Dispersion model
Bint_sd <- as_draws_matrix(d_yr_area, variable = "b_sigma_Intercept")
# Convert to variance scale scale
# brms uses residual standard deviation as the response variable for the dispersion model (ln(sd)).
# We need to convert to variance scale by multiplying by 2 for the dispersion model intercepts.
```

```
B_pv <- 2 * Bint_sd

## Extract the random ID intercept variance components.
# Mean model
sigma_ID_m0 <- as_draws_matrix(d_yr_area, variable = "sd_id__Intercept")
# Dispersion model
sigma_ID_sd0 <- as_draws_matrix(d_yr_area, variable = "sd_id__sigma_Intercept")
# Convert to variance scale scale
sigma_ID_v0 <- 2 * sigma_ID_sd0

## Repeat process for the random deploy year intercept variance components.
# Mean model
sigma_deploy.yr_m0 <- as_draws_matrix(d_yr_area, variable = "sd_deploy.yr__Intercept")
# Dispersion model
sigma_deploy.yr_sd0 <- as_draws_matrix(d_yr_area, variable = "sd_deploy.yr__sigma_Intercept")
# Convert to variance scale scale
sigma_deploy.yr_v0 <- 2 * sigma_deploy.yr_sd0

## Calculate the fixed effect variance for the mean model (var_fixed_m) and dispersion model (var_fixed_v).
# Mean model
fixed_pred_m <- posterior_linpred(d_yr_area,
                                  newdata = hr.data,
                                  re.form = NA)

var_fixed_m <- sapply(1:length(fixed_pred_m[,1]), function(x) var(fixed_pred_m[x,]))

# Dispersion model
# Intercept only for sigma so no variance for fixed effects
var_fixed_sd <- 0
# Convert to variance scale
# Again, brms uses residual standard deviation as the response variable for the dispersion model (ln(sd)).
# Variance on the ln(sd) scale can be converted to variance on the ln(var) scale by multipling by 4.
var_fixed_v <- 4 * var_fixed_sd

####################################################################################################################
### Mean Model - Repeatability and coefficient of variation ########################################################
####################################################################################################################
## Calculate average residual variance
# summed variance components
var_within_exp <- func_sum_var(vars = data.frame(sigma_ID_v0^2,
                                                 sigma_deploy.yr_v0^2))
# conversion back from log scale
sigma_w <- exp(B_pv + var_within_exp/2)

## Calculate total phenotypic variance
# Total phenotypic variance (with fixed effects)
var_p <- func_sum_var(vars = data.frame(sigma_ID_m0^2,
                                        sigma_deploy.yr_m0^2,
                                        var_fixed_m,
                                        sigma_w^2))

# 'Adjusted' phenotypic variance (without fixed effects)
var_p_nofixef <- func_sum_var(vars = data.frame(sigma_ID_m0^2,
                                                sigma_deploy.yr_m0^2,
                                                sigma_w^2))

## Calculate Repeatability
# Agreement repeatability (fixed effects var in denominator)
Rp_mu_area <- sigma_ID_m0^2/var_p

# Adjusted repeatability (fixed effects var excluded from denominator)
Rp_mu_adj_area <- sigma_ID_m0^2/var_p_nofixef

## Calculate Coefficient of variation
CV_mu_area <- sigma_ID_m0/B_pm

####################################################################################################################
```

```
### Dispersion Model - Repeatability and coefficient of variation ##################################################
###################################################################################################################
## Total variance in residual variance
# On log-normal scale:
var_residvar_exp <- func_sum_var(vars = data.frame(sigma_ID_v0^2,
                                                   sigma_deploy.yr_v0^2))
# Converting back to same scale as mean model:
var_residvar <- (exp(var_residvar_exp) - 1) * exp(2 * B_pv + var_residvar_exp)

## Total variance in phenotypic variance (repeatability denominator)
total_var_var <- 2 * var_p^2 + 3 * var_residvar

## Getting variance in individual component (repeatability numerator) through the
## preservation of the proportionality (i.e. ratio method; O'dea et al. 2022)
var_ID <- var_residvar * (sigma_ID_v0^2/var_residvar_exp)

## Calculate repeatability for the dispersion model
Rp_var_area <- var_ID/total_var_var

## Calculate coefficient of variation for the dispersion model
CV_var_area <- sqrt(var_ID)/sigma_w^2
```

Shape index

Model equation

$$shapeindex_{ijk} \;\sim\; Lognormal\left(\mathrm{y}_{ijk}, \sigma^2\right)$$

$$log\left(\mathrm{y}_{ijk}\right) = \left(\beta_{m0} + ID_{m0j} + Yr_{m0k}\right) + \beta_{n.loc_{ij}}\sqrt{x_{n.loc_{ij}}} + \beta_{\mathrm{period}_i}x_{\mathrm{period}_i} +$$

$$\beta_{\mathrm{sexage}_j}x_{\mathrm{sexage}_j} + \beta_{\mathrm{sexage:period}_{ij}}x_{\mathrm{sexage:period}_{ij}} + f_{\mathrm{period}_i}\left(x_{yr_k}\right) + \epsilon_{ijk}$$

$$\ln\left(\sigma_{\epsilon_{ijk}}\right) = \left(\beta_{sd0_{exp}} + ID_{sd0j_{exp}} + Yr_{sd0k_{exp}}\right)$$

$$\epsilon_{ijk} \;\sim\; Normal\left(0, \sigma^2_{\epsilon_{ijk}}\right)$$

$$\begin{bmatrix} ID_{m0j} \\ Yr_{m0k} \\ ID_{sd0j_{exp}} \\ Yr_{sd0k_{exp}} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{ID_{m0}} & \dots & \rho(ID_{m0j}, ID_{sd0j_{exp}})\sigma_{ID_{m0}}\sigma_{ID_{sd0_{exp}}} & \dots \\ \dots & \sigma^2_{Yr_{m0}} & \dots & \rho(Yr_{m0j}, Yr_{sd0j_{exp}})\sigma_{Yr_{m0}}\sigma_{Yr_{sd0_{exp}}} \\ \dots & \dots & \sigma^2_{ID_{sd0_{exp}}} & \dots \\ \dots & \dots & \dots & \sigma^2_{Yr_{sd0}} \end{bmatrix}\right)$$

where, y_{ijk} is the shape index of the home range for the ith month of the jth individual in the kth year, β_{m0} is the population intercept of the mean model, ID_{m0j} is the difference between the population intercept and the random intercept for the jth individual for the mean model, Yr_{m0k} is the difference between the population intercept and the random intercept for the kth year of deployment for the mean model, $\beta_{n.loc_{ij}}$ is the population slope of the number of locations used to estimate the home range ($x_{n.loc_{ij}}$) of individual j for month j, β_{period_i} is the population slope of a categorical input variable of the period of the year (x_{period_i}) corresponding to month i for the mean model, β_{sexage_j} is the population slope of a categorical input variable of the sex-age class (x_{sexage_j}) of individual j for the mean model, $\beta_{\mathrm{sexage:period}_{ij}}$ is the population slope of the interaction between period of the year and sex-age class ($x_{\mathrm{sexage:period}_{ij}}$) of individual j in month j for the mean model, $f_{\mathrm{period}_i}\left(x_{yr_k}\right)$ is the population slope of year of deployment k (x_{yr_k}) for the period of the year corresponding to month i modelled as a thin plate regression spline for the mean model, ϵ_{ijk}

is the residual error for the ith month of the jth individual in the kth year, $\sigma_{\epsilon_{ijk}}$ and $\sigma^2_{\epsilon_{ijk}}$ are the residual standard deviation and variance for the ith month of the jth individual in the kth year,

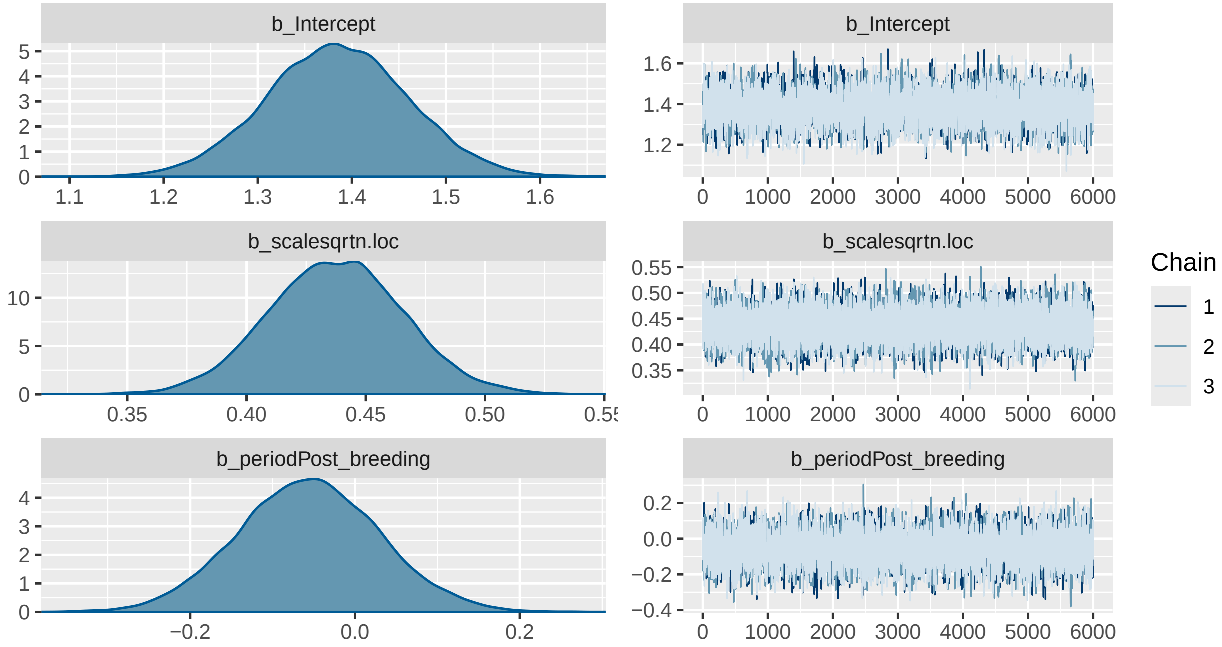
$\beta_{sd0_{exp}}$ is the population intercept of the dispersion model estimated on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $ID_{sd0j_{exp}}$ is the difference between the population intercept and the random intercept for the jth individual for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $Yr_{sd0k_{exp}}$ is the difference between the population intercept and the random intercept for the kth year of deployment for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale,

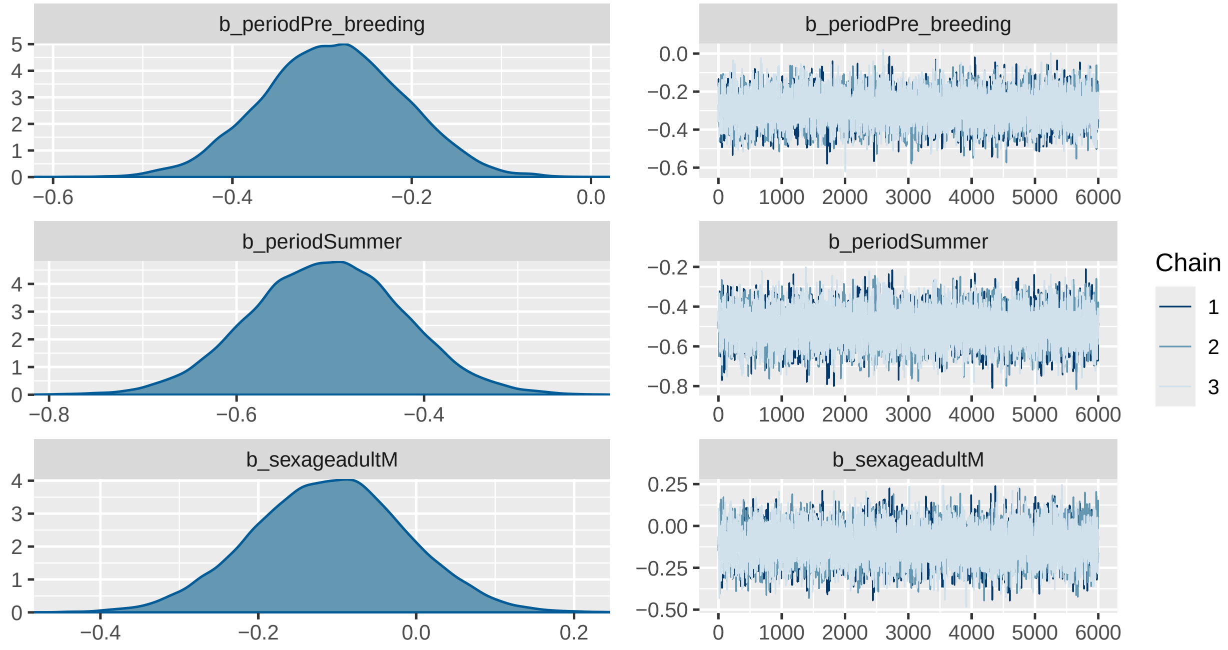
MVN is a multivariate normal distribution, $\sigma^2_{ID_{m0}}$ is the between-individual variance for the individual intercepts for the mean model, $\sigma^2_{Yr_{m0}}$ is the between-year variance for the year intercepts for the mean model, $\sigma^2_{ID_{sd0_{exp}}}$ is the between-individual variance for the individual intercepts for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $\sigma^2_{Yr_{sd0}}$ is the between-year variance for the year intercepts for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $\rho(ID_{m0j}, ID_{sd0j_{exp}})\sigma_{ID_{m0}}\sigma_{ID_{sd0_{exp}}}$ is the covariance between individual intercepts in the mean and dispersion models written as the product of the correlation between the intercepts ($\rho(ID_{m0j}, ID_{sd0j_{exp}})$ and the standard deviation for the intercepts of the mean ($\sigma_{ID_{m0}}$) and dispersion ($\sigma_{ID_{sd0_{exp}}}$) models to illustrate how the model is estimates the correlation terms, $\rho(Yr_{m0j}, Yr_{sd0j_{exp}})\sigma_{Yr_{m0}}\sigma_{Yr_{sd0_{exp}}}$ is the covariance between year intercepts in the mean and dispersion models.

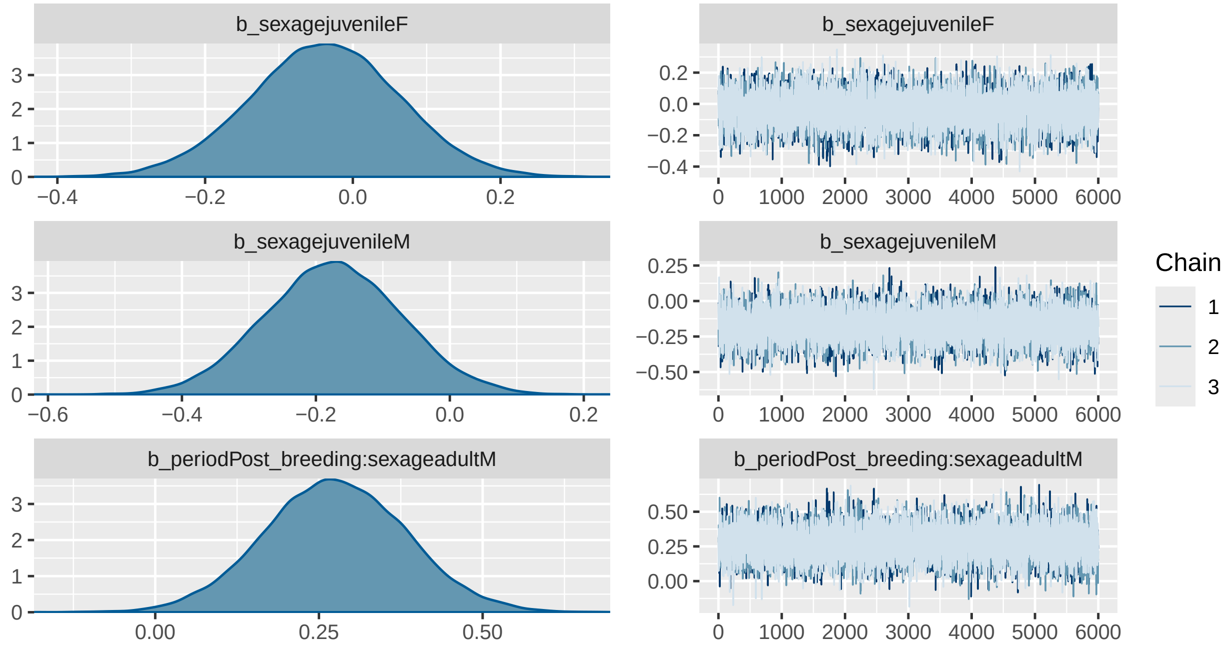
Model code

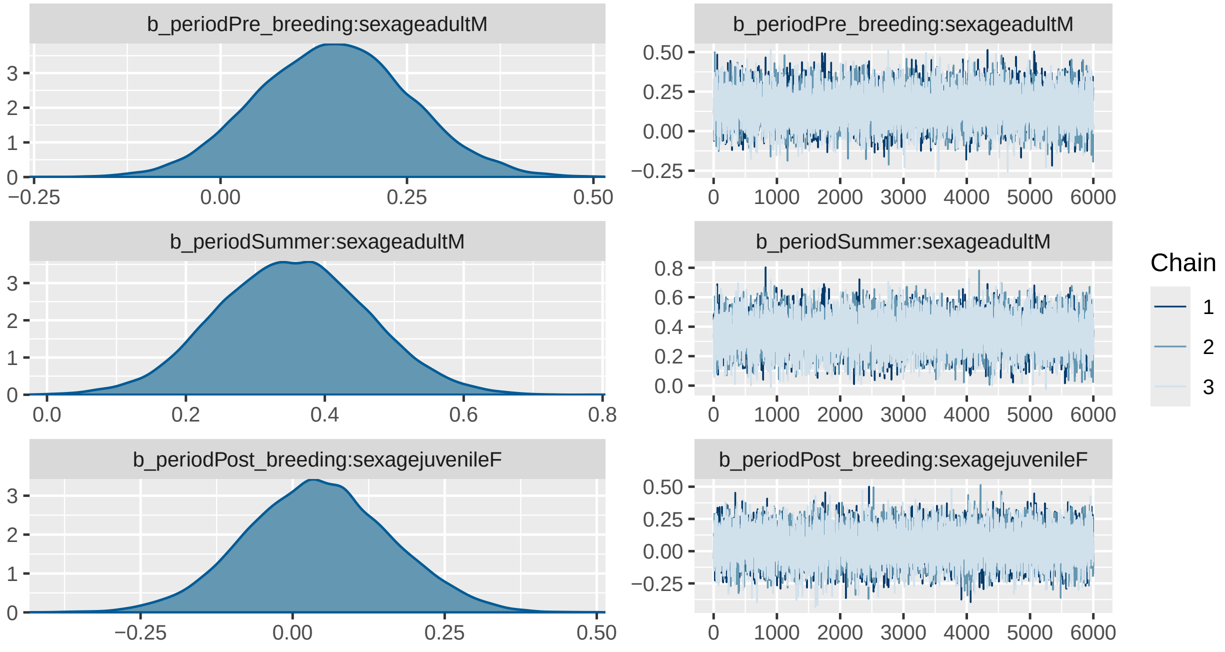
```
d_yr_shape_form <- bf(
  shapeindex ~ 0 + Intercept + scale(sqrt(n.loc)) +
    period + sexage + sexage:period +
    s(scale(deploy.yr), by = period) +
    (1|p|id) + (1|y|deploy.yr),
  sigma ~ 0 + Intercept + (1|p|id) + (1|y|deploy.yr),
  family = brmsfamily("lognormal"))

d_yr_shape_prior <- c(

  prior(lkj(1), class = "cor", group = "id"),

  prior(normal(2.3, 0.7), class = "b", coef = "Intercept"),

  prior(normal(0.5, 0.5), class = "b", coef = "scalesqrtn.loc"),
  prior(normal(0, 0.7), class = "b", coef = "sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding:sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding:sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "periodPost_breeding:sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding:sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding:sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "periodPre_breeding:sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer:sexageadultM"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer:sexagejuvenileF"),
  prior(normal(0, 0.7), class = "b", coef = "periodSummer:sexagejuvenileM"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodBreeding_1"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodPost_breeding_1"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodPre_breeding_1"),
  prior(normal(0, 0.7), class = "b", coef = "sscaledeploy.yr:periodSummer_1"),

  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "id"),
  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "deploy.yr"),

  prior(normal(0, 0.7), class = "b", coef = "Intercept", dpar = "sigma"),

  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "deploy.yr", dpar = "sigma"),
  prior(normal(0, 1), class = "sd", coef = "Intercept", group = "id", dpar = "sigma"),

  prior(normal(0, 1), class = "sds", coef = "s(scale(deploy.yr), by = period)")

)

d_yr_shape <- brm(
  d_yr_shape_form,
  prior = d_yr_shape_prior,
  data = hr.data,
  save_pars = save_pars(all = TRUE),
  sample_prior = TRUE,
  iter = 8000, cores = 3, chains = 3, warmup = 2000, thin = 1,
  control = list(adapt_delta = 0.99),
  file = "Objects/models/shape.rds",
  file_refit = "on_change")
```

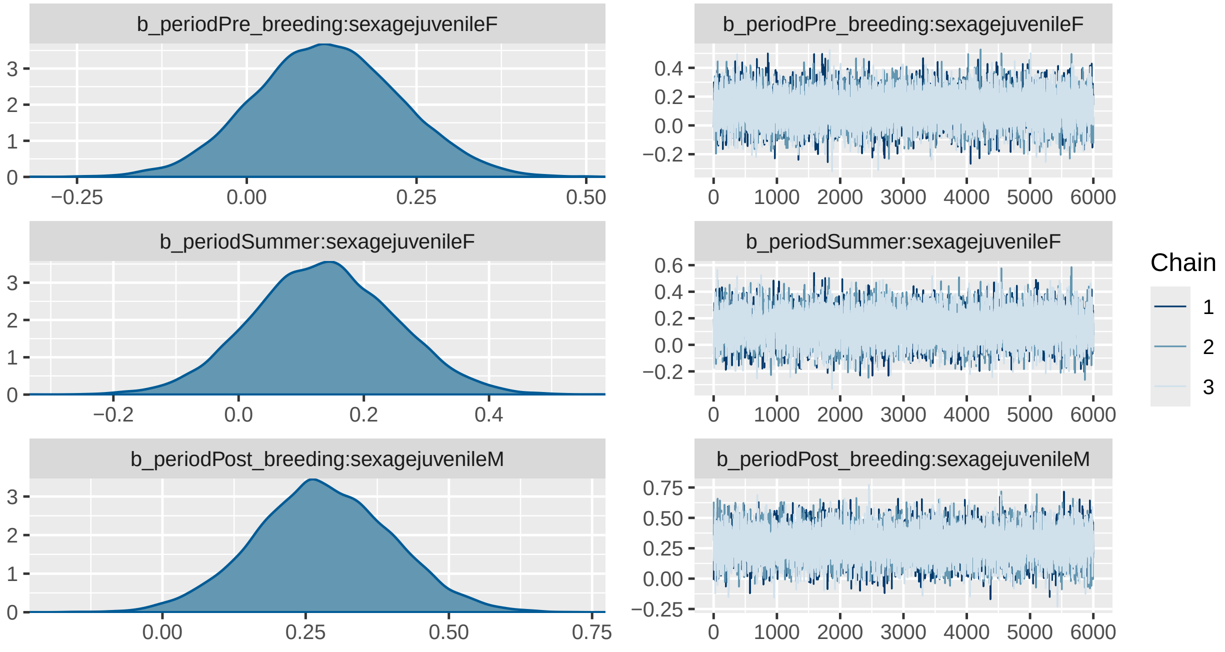
Convergence diagnostic

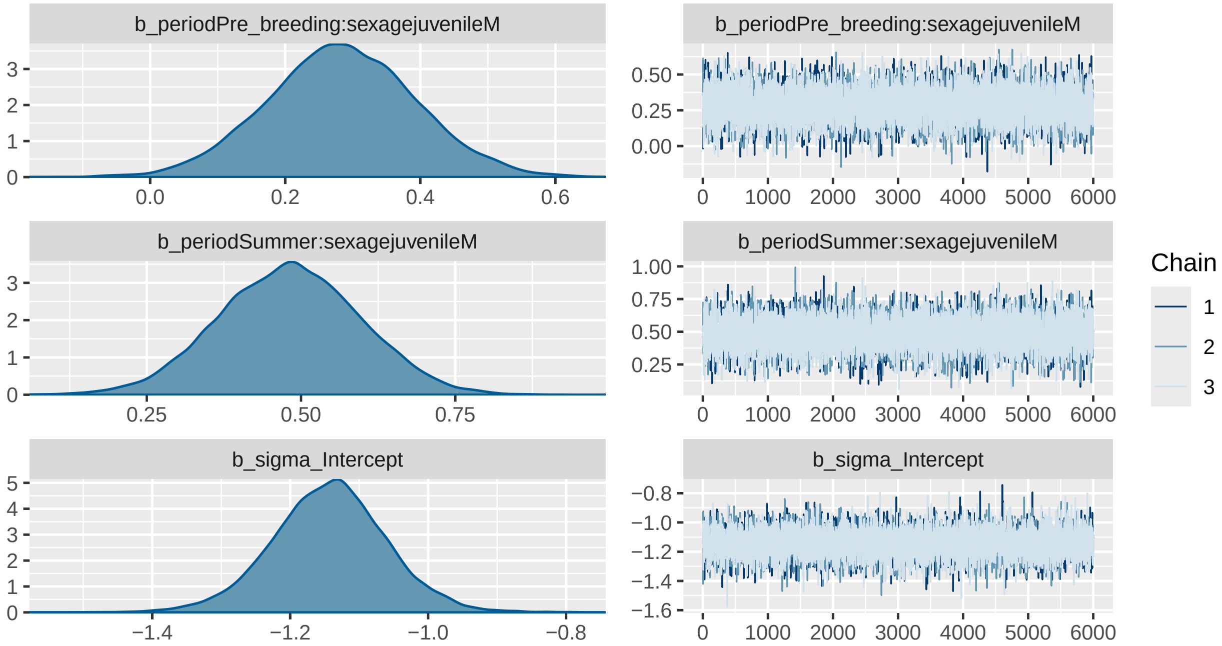


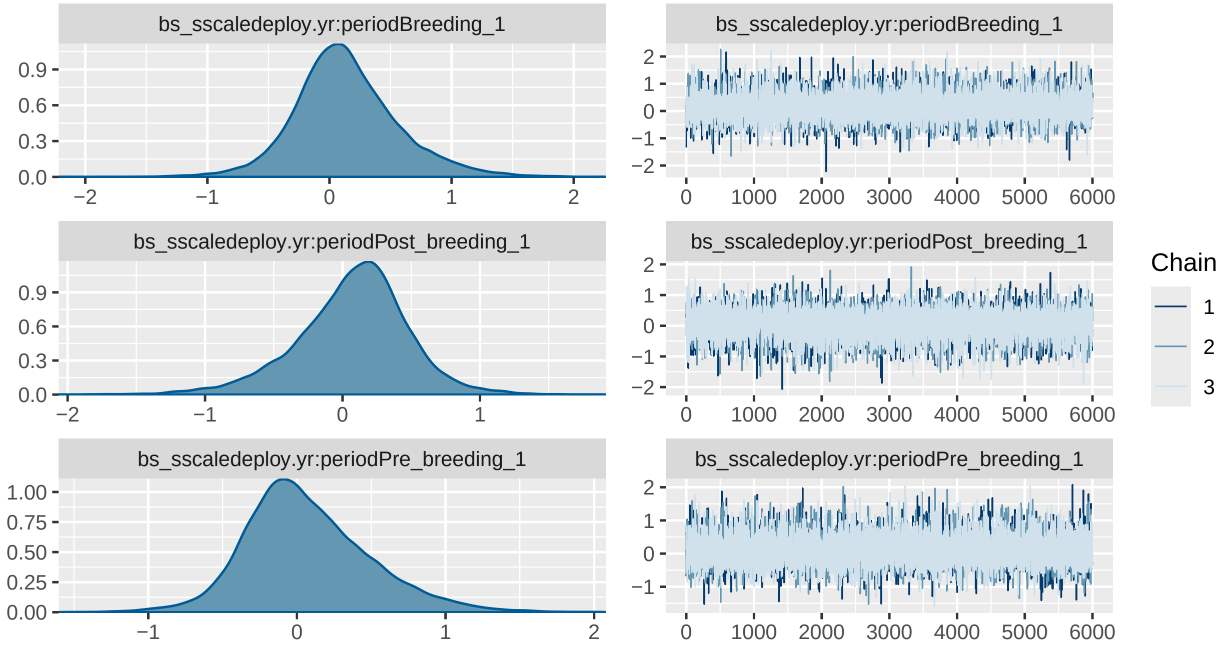

b_periodPre_breeding
5
4
3
2
1
0
−0.6
−0.4
−0.2
0.0
b_periodPre_breeding
0.0
−0.2
−0.4
−0.6
0
1000
2000
3000
4000
5000
6000
b_periodSummer
4
3
2
1
0
−0.8
−0.6
−0.4
b_periodSummer
−0.2
−0.4
−0.6
−0.8
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_sexageadultM
4
3
2
1
0
−0.4
−0.2
0.0
0.2
b_sexageadultM
0.25
0.00
−0.25
−0.50
0
1000
2000
3000
4000
5000
6000

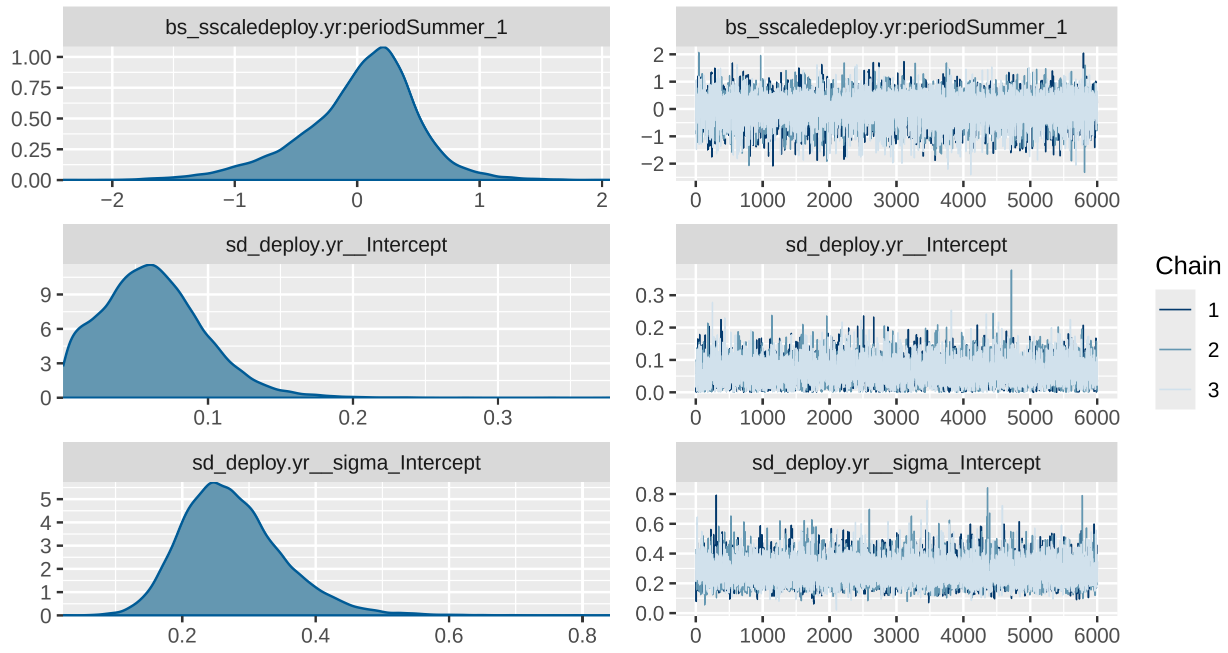

b_sexagejuvenileF
3
2
1
0
−0.4
−0.2
0.0
0.2
b_sexagejuvenileF
0.2
0.0
−0.2
−0.4
0
1000
2000
3000
4000
5000
6000
b_sexagejuvenileM
3
2
1
0
−0.6
−0.4
−0.2
0.0
0.2
b_sexagejuvenileM
0.25
0.00
−0.25
−0.50
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_periodPost_breeding:sexageadultM
3
2
1
0
0.00
0.25
0.50
b_periodPost_breeding:sexageadultM
0.50
0.25
0.00
0
1000
2000
3000
4000
5000
6000

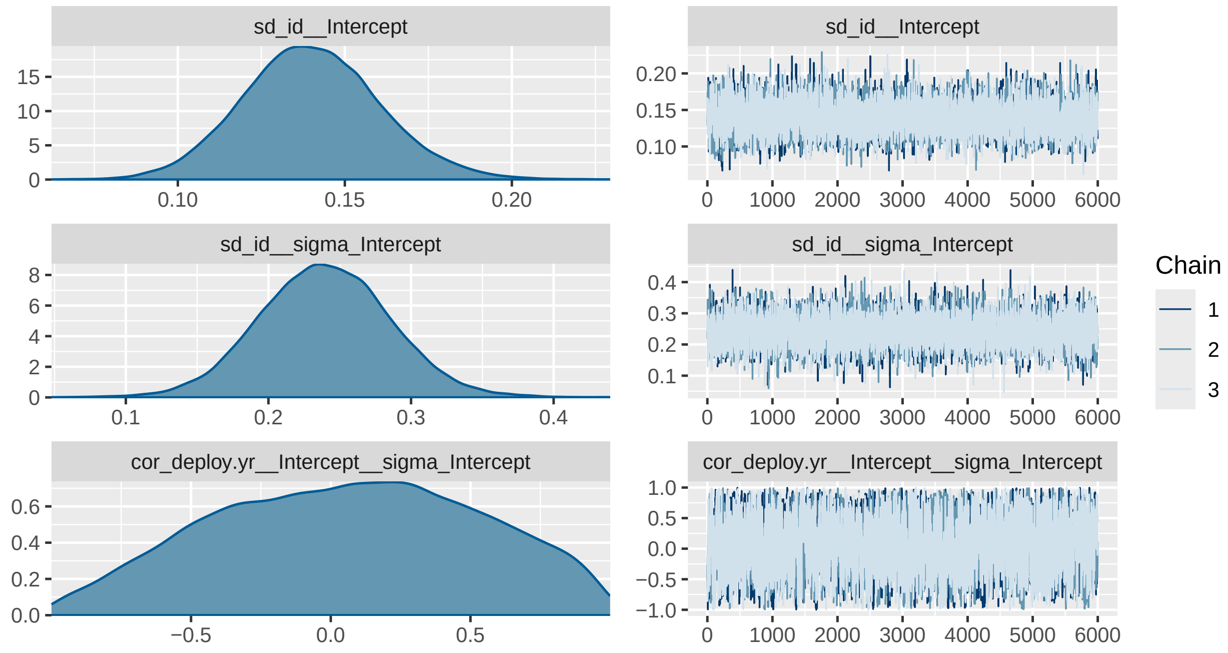

b_periodPre_breeding:sexageadultM
3
2
1
0
−0.25
0.00
0.25
0.50
b_periodPre_breeding:sexageadultM
0.50
0.25
0.00
−0.25
0
1000
2000
3000
4000
5000
6000
b_periodSummer:sexageadultM
3
2
1
0
0.0
0.2
0.4
0.6
0.8
b_periodSummer:sexageadultM
0.8
0.6
0.4
0.2
0.0
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_periodPost_breeding:sexagejuvenileF
3
2
1
0
−0.25
0.00
0.25
0.50
b_periodPost_breeding:sexagejuvenileF
0.50
0.25
0.00
−0.25
0
1000
2000
3000
4000
5000
6000

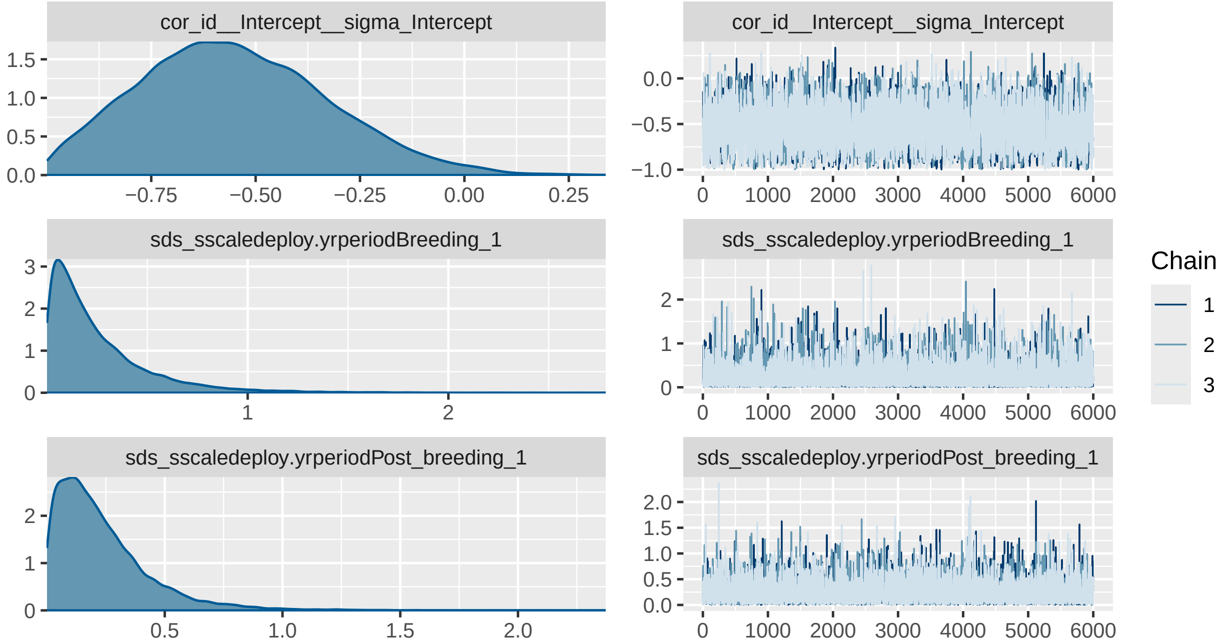

b_periodPre_breeding:sexagejuvenileF
3
2
1
0
−0.25
0.00
0.25
0.50
b_periodPre_breeding:sexagejuvenileF
0.4
0.2
0.0
−0.2
0
1000
2000
3000
4000
5000
6000
b_periodSummer:sexagejuvenileF
3
2
1
0
−0.2
0.0
0.2
0.4
b_periodSummer:sexagejuvenileF
0.6
0.4
0.2
0.0
−0.2
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_periodPost_breeding:sexagejuvenileM
3
2
1
0
0.00
0.25
0.50
0.75
b_periodPost_breeding:sexagejuvenileM
0.75
0.50
0.25
0.00
−0.25
0
1000
2000
3000
4000
5000
6000

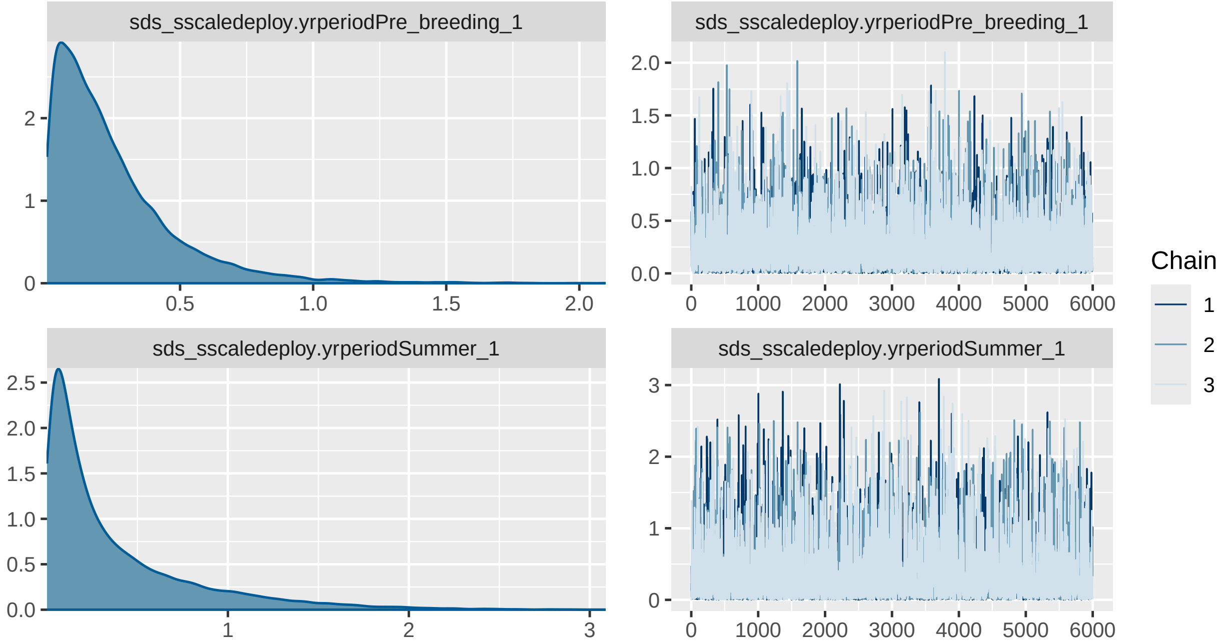

b_periodPre_breeding:sexagejuvenileM
3
2
1
0
0.0
0.2
0.4
0.6
b_periodPre_breeding:sexagejuvenileM
0.50
0.25
0.00
0
1000
2000
3000
4000
5000
6000
b_periodSummer:sexagejuvenileM
3
2
1
0
0.25
0.50
0.75
b_periodSummer:sexagejuvenileM
1.00
0.75
0.50
0.25
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
b_sigma_Intercept
5
4
3
2
1
0
−1.4
−1.2
−1.0
−0.8
b_sigma_Intercept
−0.8
−1.0
−1.2
−1.4
−1.6
0
1000
2000
3000
4000
5000
6000


bs_sscaledeploy.yr:periodBreeding_1
0.9
0.6
0.3
0.0
−2
−1
0
1
2
bs_sscaledeploy.yr:periodBreeding_1
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000
bs_sscaledeploy.yr:periodPost_breeding_1
0.9
0.6
0.3
0.0
−2
−1
0
1
bs_sscaledeploy.yr:periodPost_breeding_1
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
bs_sscaledeploy.yr:periodPre_breeding_1
1.00
0.75
0.50
0.25
0.00
−1
0
1
2
bs_sscaledeploy.yr:periodPre_breeding_1
2
1
0
−1
0
1000
2000
3000
4000
5000
6000

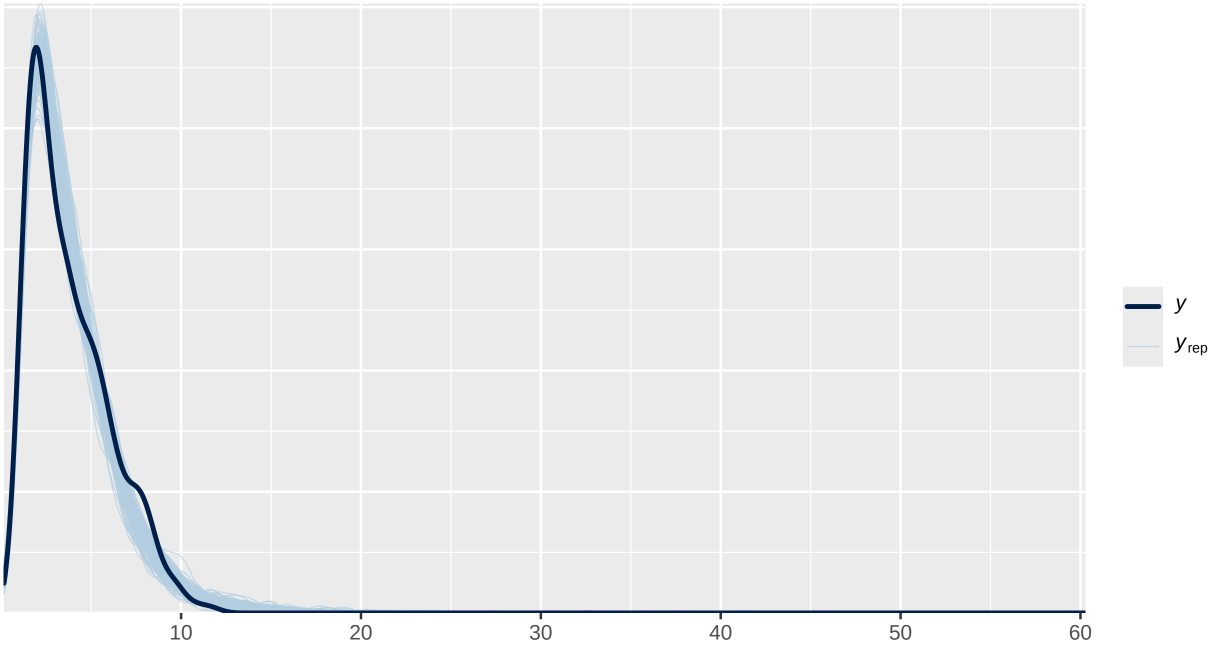

bs_sscaledeploy.yr:periodSummer_1
1.00
0.75
0.50
0.25
0.00
−2
−1
0
1
2
bs_sscaledeploy.yr:periodSummer_1
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000
sd_deploy.yr__Intercept
9
6
3
0
0.1
0.2
0.3
sd_deploy.yr__Intercept
0.3
0.2
0.1
0.0
Chain
1
2
3
sd_deploy.yr__sigma_Intercept
5
4
3
2
1
0
0.2
0.4
0.6
0.8
sd_deploy.yr__sigma_Intercept
0.8
0.6
0.4
0.2
0.0


sd_id__Intercept
15
10
5
0
0.10
0.15
0.20
sd_id__Intercept
0.20
0.15
0.10
0
1000
2000
3000
4000
5000
6000
sd_id__sigma_Intercept
8
6
4
2
0
0.1
0.2
0.3
0.4
sd_id__sigma_Intercept
0.4
0.3
0.2
0.1
Chain
1
2
3
cor_deploy.yr__Intercept__sigma_Intercept
0.6
0.4
0.2
0.0
−0.5
0.0
0.5
cor_deploy.yr__Intercept__sigma_Intercept
1.0
0.5
0.0
−0.5
−1.0


cor_id__Intercept__sigma_Intercept
1.5
1.0
0.5
0.0
−0.75
−0.50
−0.25
0.00
0.25
cor_id__Intercept__sigma_Intercept
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
6000
sds_sscaledeploy.yrperiodBreeding_1
3
2
1
0
1
2
sds_sscaledeploy.yrperiodBreeding_1
2
1
0
Chain
1
2
3
sds_sscaledeploy.yrperiodPost_breeding_1
2
1
0
0.5
1.0
1.5
2.0
sds_sscaledeploy.yrperiodPost_breeding_1
2.0
1.5
1.0
0.5
0.0

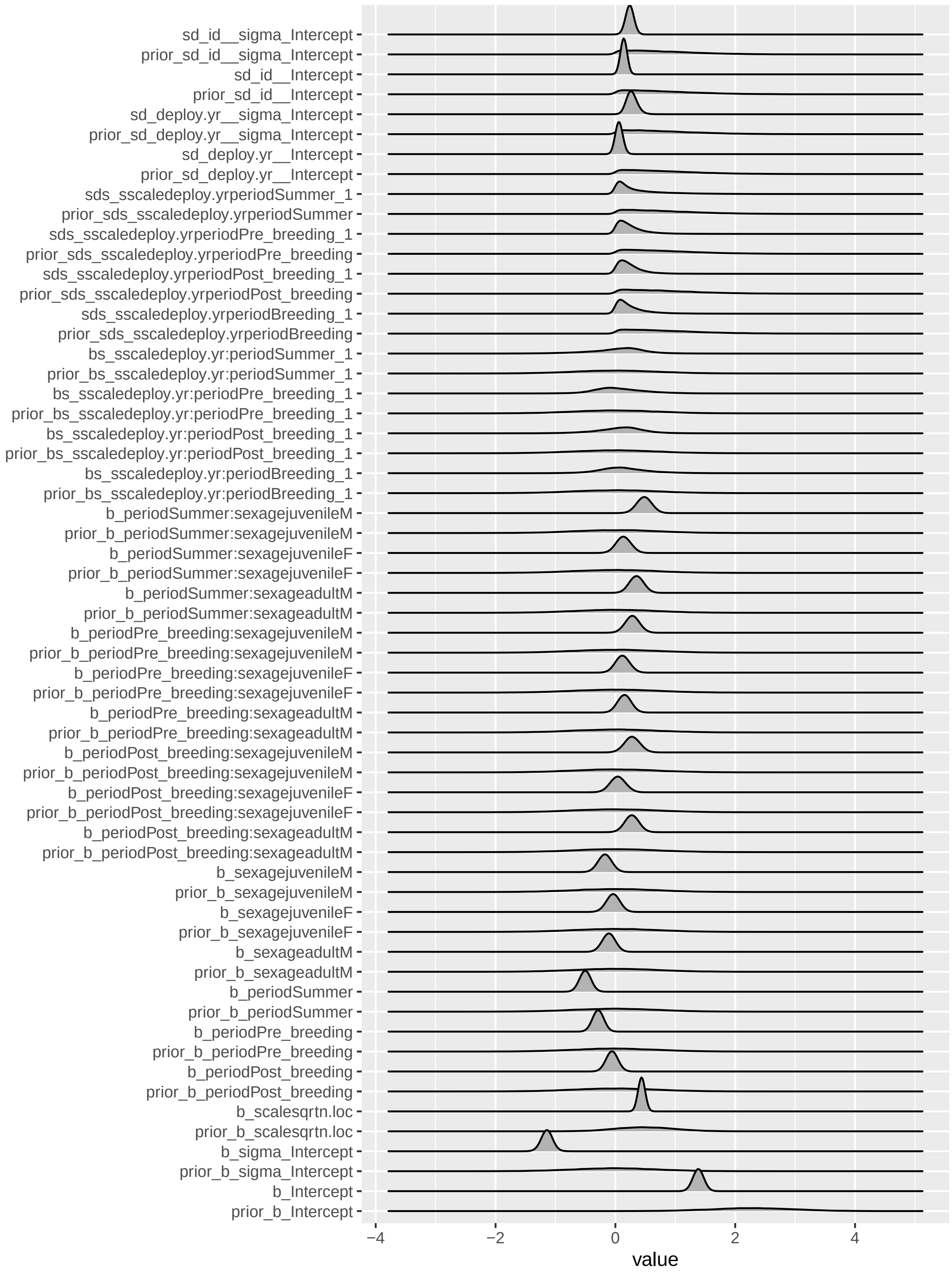

sds_sscaledeploy.yrperiodPre_breeding_1
2
1
0
0.5
1.0
1.5
2.0
sds_sscaledeploy.yrperiodPre_breeding_1
2.0
1.5
1.0
0.5
0.0
0
1000
2000
3000
4000
5000
6000
Chain
1
2
3
sds_sscaledeploy.yrperiodSummer_1
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
sds_sscaledeploy.yrperiodSummer_1
3
2
1
0

Posterior predictive checks



Priors vs posteriors

```
Picking joint bandwidth of 0.0514
```


sd_id__sigma_Intercept
prior_sd_id__sigma_Intercept
sd_id__Intercept
prior_sd_id__Intercept
sd_deploy.yr__sigma_Intercept
prior_sd_deploy.yr__sigma_Intercept
sd_deploy.yr__Intercept
prior_sd_deploy.yr__Intercept
sds_sscaledeploy.yrperiodSummer_1
prior_sds_sscaledeploy.yrperiodSummer
sds_sscaledeploy.yrperiodPre_breeding_1
prior_sds_sscaledeploy.yrperiodPre_breeding
sds_sscaledeploy.yrperiodPost_breeding_1
prior_sds_sscaledeploy.yrperiodPost_breeding
sds_sscaledeploy.yrperiodBreeding_1
prior_sds_sscaledeploy.yrperiodBreeding
bs_sscaledeploy.yr:periodSummer_1
prior_bs_sscaledeploy.yr:periodSummer_1
bs_sscaledeploy.yr:periodPre_breeding_1
prior_bs_sscaledeploy.yr:periodPre_breeding_1
bs_sscaledeploy.yr:periodPost_breeding_1
prior_bs_sscaledeploy.yr:periodPost_breeding_1
bs_sscaledeploy.yr:periodBreeding_1
prior_bs_sscaledeploy.yr:periodBreeding_1
b_periodSummer:sexagejuvenileM
prior_b_periodSummer:sexagejuvenileM
b_periodSummer:sexagejuvenileF
prior_b_periodSummer:sexagejuvenileF
b_periodSummer:sexageadultM
prior_b_periodSummer:sexageadultM
b_periodPre_breeding:sexagejuvenileM
prior_b_periodPre_breeding:sexagejuvenileM
b_periodPre_breeding:sexagejuvenileF
prior_b_periodPre_breeding:sexagejuvenileF
b_periodPre_breeding:sexageadultM
prior_b_periodPre_breeding:sexageadultM
b_periodPost_breeding:sexagejuvenileM
prior_b_periodPost_breeding:sexagejuvenileM
b_periodPost_breeding:sexagejuvenileF
prior_b_periodPost_breeding:sexagejuvenileF
b_periodPost_breeding:sexageadultM
prior_b_periodPost_breeding:sexageadultM
b_sexagejuvenileM
prior_b_sexagejuvenileM
b_sexagejuvenileF
prior_b_sexagejuvenileF
b_sexageadultM
prior_b_sexageadultM
b_periodSummer
prior_b_periodSummer
b_periodPre_breeding
prior_b_periodPre_breeding
b_periodPost_breeding
prior_b_periodPost_breeding
b_scalesqrtn.loc
prior_b_scalesqrtn.loc
b_sigma_Intercept
prior_b_sigma_Intercept
b_Intercept
prior_b_Intercept
−4
−2
0
2
4
value

Model summary

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Population-Level Effects:							
Intercept	1.385	0.075	1.237	1.533	1.000	3 248	6 463
scalesqrtn.loc	0.437	0.028	0.382	0.493	1.000	13 652	13 175
periodPost__breeding	-0.056	0.086	-0.224	0.114	1.000	3 943	7 611
periodPre__breeding	-0.288	0.078	-0.440	-0.137	1.001	3 485	7 403
periodSummer	-0.503	0.081	-0.663	-0.344	1.001	3 431	7 023
sexageadultM	-0.110	0.097	-0.301	0.079	1.000	3 600	7 147
sexagejuvenileF	-0.038	0.101	-0.239	0.159	1.000	3 847	7 417
sexagejuvenileM	-0.174	0.102	-0.374	0.028	1.000	4 169	8 308
periodPost__breeding:sexageadultM	0.275	0.108	0.063	0.488	1.000	4 645	8 882
periodPre__breeding:sexageadultM	0.152	0.102	-0.046	0.355	1.000	4 262	8 958
periodSummer:sexageadultM	0.353	0.108	0.145	0.566	1.000	4 080	8 598
periodPost__breeding:sexagejuvenileF	0.043	0.119	-0.191	0.278	1.000	4 757	9 623
periodPre__breeding:sexagejuvenileF	0.119	0.107	-0.088	0.332	1.000	4 117	8 617
periodSummer:sexagejuvenileF	0.136	0.113	-0.083	0.360	1.000	4 242	7 790
periodPost__breeding:sexagejuvenileM	0.281	0.120	0.046	0.518	1.000	5 016	9 178
periodPre__breeding:sexagejuvenileM	0.283	0.110	0.067	0.501	1.000	4 569	8 490
periodSummer:sexagejuvenileM	0.481	0.115	0.258	0.704	1.001	4 420	8 767
sigma__Intercept	-1.144	0.083	-1.311	-0.979	1.000	7 622	10 083
sscaledeploy.yr:periodBreeding__1	0.138	0.430	-0.655	1.092	1.000	10 601	11 103
sscaledeploy.yr:periodPost__breeding__1	0.072	0.407	-0.830	0.825	1.000	10 285	11 342
sscaledeploy.yr:periodPre__breeding__1	0.072	0.422	-0.656	1.018	1.000	9 100	10 201
sscaledeploy.yr:periodSummer__1	0.023	0.474	-1.077	0.855	1.000	9 674	10 268
Smooth Terms:							
sds(sscaledeploy.yrperiodBreeding__1)	0.251	0.255	0.008	0.945	1.001	6 521	9 529
sds(sscaledeploy.yrperiodPost__breeding__1)	0.235	0.201	0.010	0.758	1.000	7 412	7 140
sds(sscaledeploy.yrperiodPre__breeding__1)	0.246	0.230	0.009	0.857	1.000	6 345	8 567
sds(sscaledeploy.yrperiodSummer__1)	0.374	0.427	0.007	1.574	1.000	3 545	7 126
Group-Level Effects:							
~Id (Number of levels: 124)							
sd(Intercept__id)	0.140	0.021	0.101	0.182	1.000	5 487	9 679
sd(sigma__Intercept__id)	0.240	0.046	0.150	0.330	1.000	5 014	7 859
cor(Intercept__id,sigma__Intercept__id)	-0.551	0.223	-0.937	-0.081	1.001	2 889	3 849
~Year (Number of levels: 16)							
sd(Intercept__yr)	0.062	0.035	0.005	0.138	1.001	3 121	4 804
sd(sigma__Intercept__yr)	0.276	0.076	0.151	0.445	1.000	6 173	8 925
cor(Intercept__yr,sigma__Intercept__yr)	0.060	0.469	-0.820	0.889	1.000	1 821	2 706

Post-hoc tests

```
# 'CI': 95%-CI
# '*': The value tested against lies outside the 95%-CI.
# Evidence ratio is a Bayes factor between the hypothesis and its alternative.
#   Values greater than one indicate that evidence in favor of the point
```

```r
#   hypothesis has increased after seeing the data.

####################################################################################################################
# Post-hoc comparison between periods, within sexage classes #######################################################
####################################################################################################################
## Adult females
# Define hypotheses
v_af <- c("Summer = Breeding" = "periodSummer = 0",
          "Summer =  Pre-breeding" = "periodSummer = periodPre_breeding",
          "Summer = Post-breeding" = "periodSummer = periodPost_breeding",
          "Pre-breeding = Breeding" = "periodPre_breeding = 0",
          "Pre-breeding = Post-breeding" = "periodPre_breeding = periodPost_breeding",
          "Breeding = Post-breeding" = "0 = periodPost_breeding")
# Test hypotheses
h_af <- hypothesis(d_yr_shape, v_af)
# Store results
h_af_tib <- h_af$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Adult males
# Define hypotheses
v_am <- c("Summer = Breeding" =
            "periodSummer + periodSummer:sexageadultM = 0",
          "Summer =  Pre-breeding" =
            "periodSummer + periodSummer:sexageadultM = periodPre_breeding + periodPre_breeding:sexageadultM",
          "Summer = Post-breeding" =
            "periodSummer + periodSummer:sexageadultM = periodPost_breeding + periodPost_breeding:sexageadultM",
          "Pre-breeding = Breeding" =
            "periodPre_breeding + periodPre_breeding:sexageadultM = 0",
          "Pre-breeding = Post-breeding" =
            "periodPre_breeding + periodPre_breeding:sexageadultM = periodPost_breeding +
             periodPost_breeding:sexageadultM",
          "Breeding = Post-breeding" =
            "0 = periodPost_breeding + periodPost_breeding:sexageadultM")
# Test hypotheses
h_am <- hypothesis(d_yr_shape, v_am)
# Store results
h_am_tib <- h_am$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile females
# Define hypotheses
v_jf <- c("Summer = Breeding" =
            "periodSummer + periodSummer:sexagejuvenileF = 0",
          "Summer =  Pre-breeding" =
            "periodSummer + periodSummer:sexagejuvenileF = periodPre_breeding + periodPre_breeding:sexagejuvenileF",
          "Summer = Post-breeding" =
            "periodSummer + periodSummer:sexagejuvenileF = periodPost_breeding + periodPost_breeding:sexagejuvenileF",
          "Pre-breeding = Breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileF = 0",
          "Pre-breeding = Post-breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileF = periodPost_breeding +
             periodPost_breeding:sexagejuvenileF",
          "Breeding = Post-breeding" =
            "0 = periodPost_breeding + periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_jf <- hypothesis(d_yr_shape, v_jf)
# Store results
h_jf_tib <- h_jf$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile males
# Define hypotheses
v_jm <- c("Summer = Breeding" =
            "periodSummer + periodSummer:sexagejuvenileM = 0",
          "Summer =  Pre-breeding" =
            "periodSummer + periodSummer:sexagejuvenileM = periodPre_breeding + periodPre_breeding:sexagejuvenileM",
          "Summer = Post-breeding" =
            "periodSummer + periodSummer:sexagejuvenileM = periodPost_breeding + periodPost_breeding:sexagejuvenileM",
          "Pre-breeding = Breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileM = 0",
          "Pre-breeding = Post-breeding" =
            "periodPre_breeding + periodPre_breeding:sexagejuvenileM = periodPost_breeding +
             periodPost_breeding:sexagejuvenileM",
          "Breeding = Post-breeding" =
            "0 = periodPost_breeding + periodPost_breeding:sexagejuvenileM")
# Test hypotheses
h_jm <- hypothesis(d_yr_shape, v_jm)
# Store results
h_jm_tib <- h_jm$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_shape_seas <-
  full_join(h_af_tib, h_am_tib, by = "Hypothesis") %>%
  full_join(., h_jf_tib, by = "Hypothesis") %>%
  full_join(., h_jm_tib, by = "Hypothesis")

####################################################################################################################
# Post-hoc comparison between sexage classes, within periods #######################################################
####################################################################################################################
## Summer
# Define hypotheses
v_sum <- c("Adult F = Adult M" =
             "0 = sexageadultM + periodSummer:sexageadultM",
          "Adult F = Juvenile F" =
            "0 = sexagejuvenileF + periodSummer:sexagejuvenileF",
          "Adult F = Juvenile M" =
            "0 = sexagejuvenileM + periodSummer:sexagejuvenileM",
          "Adult M = Juvenile M" =
            "sexageadultM + periodSummer:sexageadultM = sexagejuvenileM + periodSummer:sexagejuvenileM",
          "Adult M = Juvenile F" =
            "sexageadultM + periodSummer:sexageadultM = sexagejuvenileF + periodSummer:sexagejuvenileF",
          "Juvenile M = Juvenile F" =
            "sexagejuvenileM + periodSummer:sexagejuvenileM = sexagejuvenileF + periodSummer:sexagejuvenileF")
# Test hypotheses
h_sum <- hypothesis(d_yr_shape, v_sum)
# Store results
h_sum_tib <- h_sum$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Pre-breeding
# Define hypotheses
v_pre <- c("Adult F = Adult M" =
             "0 = sexageadultM + periodPre_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = sexagejuvenileF + periodPre_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = sexagejuvenileM + periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "sexageadultM + periodPre_breeding:sexageadultM = sexagejuvenileM + periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "sexageadultM + periodPre_breeding:sexageadultM = sexagejuvenileF + periodPre_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "sexagejuvenileM + periodPre_breeding:sexagejuvenileM = sexagejuvenileF +
              periodPre_breeding:sexagejuvenileF")
# Test hypotheses
h_pre <- hypothesis(d_yr_shape, v_pre)
# Store results
h_pre_tib <- h_pre$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Breeding
# Define hypotheses
v_bred <- c("Adult F = Adult M" = "0 = sexageadultM ",
           "Adult F = Juvenile F" = "0 = sexagejuvenileF",
           "Adult F = Juvenile M" = "0 = sexagejuvenileM",
           "Adult M = Juvenile M" = "sexageadultM = sexagejuvenileM",
           "Adult M = Juvenile F" = "sexageadultM = sexagejuvenileF",
           "Juvenile M = Juvenile F" = "sexagejuvenileM = sexagejuvenileF")
# Test hypotheses
h_bred <- hypothesis(d_yr_shape, v_bred)
# Store results
h_bred_tib <- h_bred$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Post-breeding
# Define hypotheses
v_post <- c("Adult F = Adult M" =
              "0 = sexageadultM + periodPost_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = sexagejuvenileF + periodPost_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = sexagejuvenileM + periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "sexageadultM + periodPost_breeding:sexageadultM = sexagejuvenileM + periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "sexageadultM + periodPost_breeding:sexageadultM = sexagejuvenileF + periodPost_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "sexagejuvenileM + periodPost_breeding:sexagejuvenileM = sexagejuvenileF +
              periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_post <- hypothesis(d_yr_shape, v_post)
# Store results
h_post_tib <- h_post$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_shape_onto <-
  full_join(h_sum_tib, h_pre_tib, by = "Hypothesis") %>%
  full_join(., h_bred_tib, by = "Hypothesis") %>%
  full_join(., h_post_tib, by = "Hypothesis")
```

Individual differences

```
####################################################################################################################
#### Extract parameters from Mean and Dispersion models ############################################################
####################################################################################################################
## Extract the population coefficients for mean model
# Adult females
Baf_data <- tibble(n.loc = mean(hr.data$n.loc),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "adult F",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Baf_all <- posterior_linpred(d_yr_shape, newdata = Baf_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Baf <- Baf_all[,1]*(4/11) + Baf_all[,2]*(3/11) + Baf_all[,3]*(1/11) +
  Baf_all[,4]*(3/11)

# Adult males
Bam_data <- tibble(n.loc = mean(hr.data$n.loc),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "adult M",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Bam_all <- posterior_linpred(d_yr_shape, newdata = Bam_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bam <- Bam_all[,1]*(4/11) + Bam_all[,2]*(3/11) + Bam_all[,3]*(1/11) +
  Bam_all[,4]*(3/11)

# Juvenile females
Bjf_data <- tibble(n.loc = mean(hr.data$n.loc),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "juvenile F",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Bjf_all <- posterior_linpred(d_yr_shape, newdata = Bjf_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bjf <- Bjf_all[,1]*(4/11) + Bjf_all[,2]*(3/11) + Bjf_all[,3]*(1/11) +
  Bjf_all[,4]*(3/11)

# Juvenile males
Bjm_data <- tibble(n.loc = mean(hr.data$n.loc),
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "juvenile M",
                   deploy.yr = mean(hr.data$deploy.yr))
# Extract the posterior draws of the linear predictor
Bjm_all <- posterior_linpred(d_yr_shape, newdata = Bjm_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bjm <- Bjm_all[,1]*(4/11) + Bjm_all[,2]*(3/11) + Bjm_all[,3]*(1/11) +
  Bjm_all[,4]*(3/11)

## Calculate the population intercept (average individual phenotype)
## for the mean model (B_pm) and the dispersion model (B_pv)
# Mean model (assuming equal representation of each sexage class)
B_pm <- Baf*(1/4) + Bam*(1/4) + Bjf*(1/4) + Bjm*(1/4)
# Dispersion model
Bint_sd <- as_draws_matrix(d_yr_shape, variable = "b_sigma_Intercept")
# Convert to variance scale scale
# brms uses residual standard deviation as the response variable for the dispersion model (ln(sd)).
# We need to convert to variance scale by multiplying by 2 for the dispersion model intercepts.
B_pv <- 2 * Bint_sd

## Extract the random ID intercept variance components.
# Mean model
sigma_ID_m0 <- as_draws_matrix(d_yr_shape, variable = "sd_id__Intercept")
# Dispersion model
sigma_ID_sd0 <- as_draws_matrix(d_yr_shape, variable = "sd_id__sigma_Intercept")
```

```
# Convert to variance scale scale
sigma_ID_v0 <- 2 * sigma_ID_sd0

## Repeat process for the random deploy year intercept variance components.
# Mean model
sigma_deploy.yr_m0 <- as_draws_matrix(d_yr_shape, variable = "sd_deploy.yr__Intercept")
# Dispersion model
sigma_deploy.yr_sd0 <- as_draws_matrix(d_yr_shape, variable = "sd_deploy.yr__sigma_Intercept")
# Convert to variance scale scale
sigma_deploy.yr_v0 <- 2 * sigma_deploy.yr_sd0

## Calculate the fixed effect variance for the mean model (var_fixed_m) and dispersion model (var_fixed_v).
# Mean model
fixed_pred_m <- posterior_linpred(d_yr_shape,
                                  newdata = hr.data,
                                  re.form = NA)

var_fixed_m <- sapply(1:length(fixed_pred_m[,1]), function(x) var(fixed_pred_m[x,]))

# Dispersion model
# Intercept only for sigma so no variance for fixed effects
var_fixed_sd <- 0
# Convert to variance scale
# Again, brms uses residual standard deviation as the response variable for the dispersion model (ln(sd)).
# Variance on the ln(sd) scale can be converted to variance on the ln(var) scale by multipling by 4.
var_fixed_v <- 4 * var_fixed_sd

####################################################################################################################
### Mean Model - Repeatability and coefficient of variation ########################################################
####################################################################################################################
## Calculate average residual variance
# summed variance components
var_within_exp <- func_sum_var(vars = data.frame(sigma_ID_v0^2,
                                                 sigma_deploy.yr_v0^2))
# conversion back from log scale
sigma_w <- exp(B_pv + var_within_exp/2)

## Calculate total phenotypic variance
# Total phenotypic variance (with fixed effects)
var_p <- func_sum_var(vars = data.frame(sigma_ID_m0^2,
                                        sigma_deploy.yr_m0^2,
                                        var_fixed_m,
                                        sigma_w^2))

# 'Adjusted' phenotypic variance (without fixed effects)
var_p_nofixef <- func_sum_var(vars = data.frame(sigma_ID_m0^2,
                                                sigma_deploy.yr_m0^2,
                                                sigma_w^2))

## Calculate Repeatability
# Agreement repeatability (fixed effects var in denominator)
Rp_mu_shape <- sigma_ID_m0^2/var_p

# Adjusted repeatability (fixed effects var excluded from denominator)
Rp_mu_adj_shape <- sigma_ID_m0^2/var_p_nofixef

## Calculate Coefficient of variation
CV_mu_shape <- sigma_ID_m0/B_pm

####################################################################################################################
### Dispersion Model - Repeatability and coefficient of variation ##################################################
####################################################################################################################
## Total variance in residual variance
# On log-normal scale:
var_residvar_exp <- func_sum_var(vars = data.frame(sigma_ID_v0^2,
                                                   sigma_deploy.yr_v0^2))
```

```
# Converting back to same scale as mean model:
var_residvar <- (exp(var_residvar_exp) - 1) * exp(2 * B_pv + var_residvar_exp)

## Total variance in phenotypic variance (repeatability denominator)
total_var_var <- 2 * var_p^2 + 3 * var_residvar

## Getting variance in individual component (repeatability numerator) through the
## preservation of the proportionality (i.e. ratio method; O'dea et al. 2022)
var_ID <- var_residvar * (sigma_ID_v0^2/var_residvar_exp)

## Calculate repeatability for the dispersion model
Rp_var_shape <- var_ID/total_var_var

## Calculate coefficient of variation for the dispersion model
CV_var_shape <- sqrt(var_ID)/sigma_w^2
```

Centroids

Model equation

$$Longitude_{ijk} \sim Normal\left(y^{Long}_{ijk}, \sigma^2\right)$$

$$Latitude_{ijk} \sim Normal\left(y^{Lat}_{ijk}, \sigma^2\right)$$

$$y^{Long}_{ijk} = \left(\beta^{Long}_{m0} + ID^{Long}_{m0j} + Yr^{Long}_{m0k}\right) + \beta^{Long}_{autocor}\, y^{Long}_{i-1jk} + \beta^{Long}_{location_j} x^{Long}_{location_j} + \beta^{Long}_{period_i} x^{Long}_{period_i} + \beta^{Long}_{sexage_j} x^{Long}_{sexage_j} + \beta^{Long}_{\text{sexage:period}_{ij}} x^{Long}_{\text{sexage:period}_{ij}} + f^{Long}_{\text{period}_i}\left(x^{Long}_{yr_k}\right) + \epsilon^{Long}_{ijk},$$

$$y^{Lat}_{ijk} = \left(\beta^{Lat}_{m0} + ID^{Lat}_{m0j} + Yr^{Lat}_{m0k}\right) + \beta^{Lat}_{autocor}\, y^{Lat}_{i-1jk} + \beta^{Lat}_{location_j} x^{Lat}_{location_j} + \beta^{Lat}_{period_i} x^{Lat}_{period_i} + \beta^{Lat}_{sexage_j} x^{Lat}_{sexage_j} + \beta^{Lat}_{\text{sexage:period}_{ij}} x^{Lat}_{\text{sexage:period}_{ij}} + f^{Lat}_{\text{period}_i}\left(x^{Lat}_{yr_k}\right) + \epsilon^{Lat}_{ijk},$$

$$\ln\left(\sigma_{\epsilon^{Long}_{ijk}}\right) = \left(\beta^{Long}_{sd0_{exp}} + ID^{Long}_{sd0j_{exp}} + Yr^{Long}_{sd0k_{exp}}\right),$$

$$\ln\left(\sigma_{\epsilon^{Lat}_{ijk}}\right) = \left(\beta^{Lat}_{sd0_{exp}} + ID^{Lat}_{sd0j_{exp}} + Yr^{Lat}_{sd0k_{exp}}\right),$$

$$\begin{bmatrix} \epsilon^{Long}_{ijk} \\ \epsilon^{Lat}_{ijk} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{\epsilon^{Long}_{ijk}} & \dots \\ \dots & \sigma^2_{\epsilon^{Long}_{ijk}} \end{bmatrix}\right),$$

$$\begin{bmatrix} ID^{Long}_{m0j} \\ Yr^{Long}_{m0k} \\ ID^{Lat}_{m0j} \\ Yr^{Lat}_{m0k} \\ ID^{Long}_{sd0j_{exp}} \\ Yr^{Long}_{sd0k_{exp}} \\ ID^{Lat}_{sd0j_{exp}} \\ Yr^{Lat}_{sd0k_{exp}} \end{bmatrix} MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{ID^{Long}_{m0j}} & \dots & \dots & \dots & \sigma_{ID^{Long}_{m0j} ID^{Long}_{sd0j_{exp}}} & \dots & \dots & \dots \\ \dots & \sigma^2_{Yr^{Long}_{m0k}} & \dots & \dots & \dots & \sigma_{Yr^{Long}_{m0k} Yr^{Long}_{sd0k_{exp}}} & \dots & \dots \\ \dots & \dots & \sigma^2_{ID^{Lat}_{m0j}} & \dots & \dots & \dots & \sigma_{ID^{Lat}_{m0j} ID^{Lat}_{sd0j_{exp}}} & \dots \\ \dots & \dots & \dots & \sigma^2_{Yr^{Lat}_{m0k}} & \dots & \dots & \dots & \sigma_{Yr^{Lat}_{m0k} Yr^{Lat}_{sd0k_{exp}}} \\ \dots & \dots & \dots & \dots & \sigma^2_{ID^{Long}_{sd0j_{exp}}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \sigma^2_{Yr^{Long}_{sd0k_{exp}}} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \sigma^2_{ID^{Lat}_{sd0j_{exp}}} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \sigma^2_{Yr^{Lat}_{sd0k_{exp}}} \end{bmatrix}\right)$$

where, y^{Long}_{ijk} and y^{Lat}_{ijk} are the coordinates of the centroid of the home range for ith month of the jth individual in the kth year, β^{Long}_{m0} and β^{Lat}_{m0} are the population intercepts of the mean model for longitude and latitude, ID^{Long}_{m0j} and ID^{Lat}_{m0j} are the differences between the population intercepts and the random intercepts for the jth individual for the mean model, Yr^{Long}_{m0k} and Yr^{Lat}_{m0k} are the differences between the population intercepts and the random intercepts for the kth year of deployment for the mean model, $\beta^{Long}_{autocor}\ y^{Long}_{i-1jk}$ and $\beta^{Lat}_{autocor}\ y^{Lat}_{i-1jk}$ are first-order autoregressive terms (AR1) grouped by individual identity for the mean model, $\beta^{Long}_{location_j}$ and $\beta^{Lat}_{location_j}$ are the population slopes of the deployment location ($x^{Long}_{location_j}$ and $x^{Lat}_{location_j}$) of individual j for month j for the mean model, $\beta^{Long}_{period_i}$ and $\beta^{Lat}_{period_i}$ are the population slopes of a categorical input variable of the period of the year ($x^{Long}_{period_i}$ and $x^{Lat}_{period_i}$) corresponding to month i for the mean model, $\beta^{Long}_{sexage_j}$ and $\beta^{Lat}_{sexage_j}$ are the population slopes of a categorical input variable of the sex-age class ($x^{Long}_{sexage_j}$ and $x^{Lat}_{sexage_j}$) of individual j for the mean model, $\beta^{Long}_{\text{sexage:period}_{ij}}$ and $\beta^{Lat}_{\text{sexage:period}_{ij}}$ are the population slopes of the interaction between period of the year and sex-age class ($x^{Long}_{\text{sexage:period}_{ij}}$ and $x^{Lat}_{\text{sexage:period}_{ij}}$) of individual j in month j for the mean model, $f^{Long}_{\text{period}_i}\left(x^{Long}_{yr_k}\right)$ and $f^{Lat}_{\text{period}_i}\left(x^{Lat}_{yr_k}\right)$ are the population slopes of year of deployment k ($x^{Long}_{yr_k}$ and $x^{Lat}_{yr_k}$) for the period of the year corresponding to month i modelled as a thin plate regression spline for the mean model, ϵ^{Long}_{ijk} and ϵ^{Lat}_{ijk} are the residual errors for the ith month of the jth individual in the kth year, $\sigma_{\epsilon^{Long}_{ijk}}$ and $\sigma_{\epsilon^{Lat}_{ijk}}$ are the residual standard deviations and $\sigma^2_{\epsilon^{Long}_{ijk}}$ and $\sigma^2_{\epsilon^{Long}_{ijk}}$ are the residual variances for the ith month of the jth individual in the kth year,

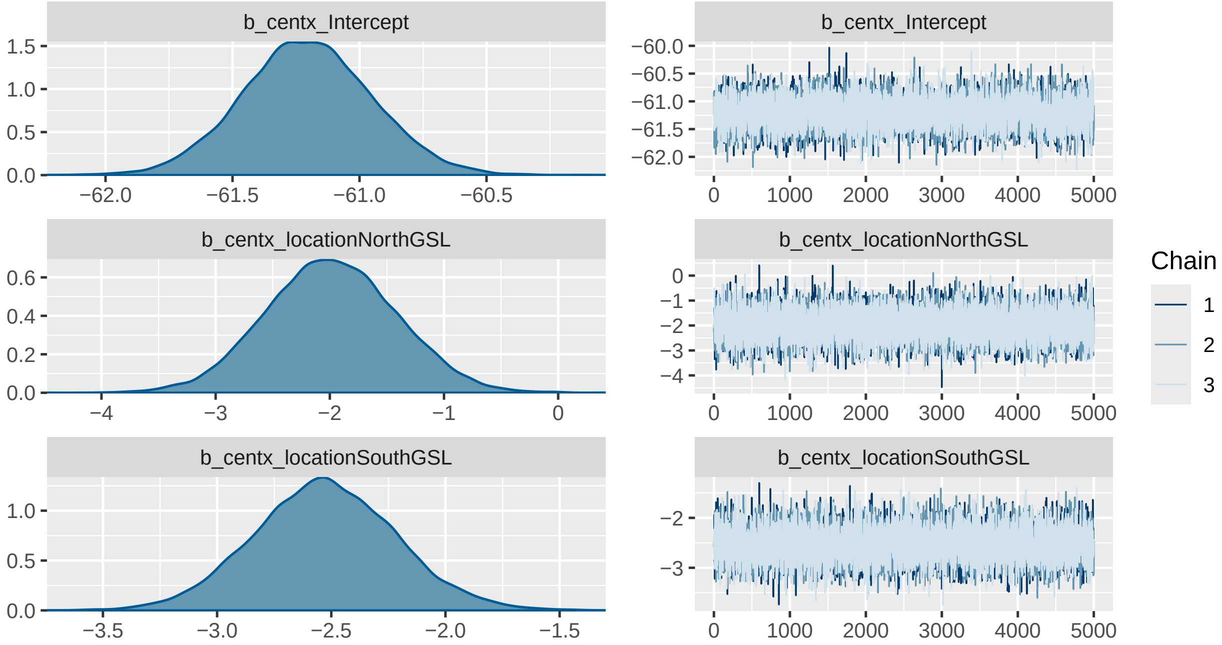
$\beta^{Long}_{sd0_{exp}}$ and $\beta^{Lat}_{sd0_{exp}}$ are the population intercepts of the dispersion model estimated on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $ID^{Long}_{sd0j_{exp}}$ and $ID^{Lat}_{sd0j_{exp}}$ are the differences between the population intercepts and the random intercepts for the jth individual for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $Yr^{Long}_{sd0k_{exp}}$ and $Yr^{Lat}_{sd0k_{exp}}$ are the differences between the population intercepts and the random intercepts for the kth year of deployment for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale,

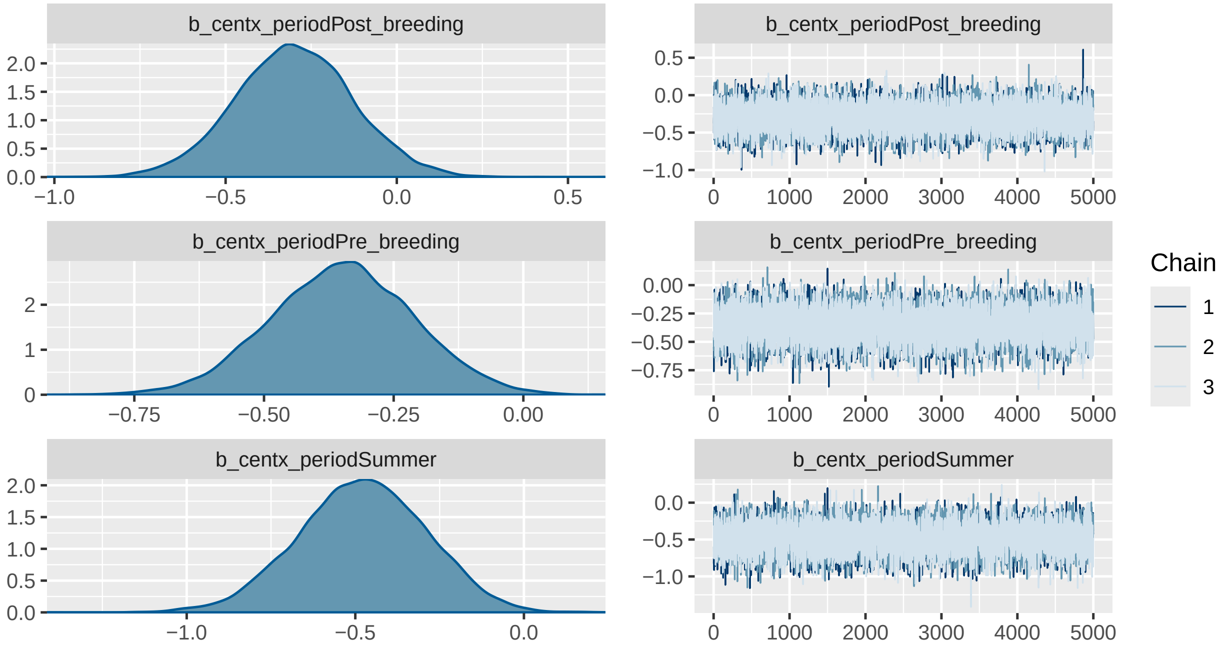
MVN is a multivariate normal distribution, $\sigma^2_{ID^{Long}_{m0j}}$ and $\sigma^2_{ID^{Lat}_{m0j}}$ are the between-individual variances for the individual intercepts for the mean model, $\sigma^2_{Yr^{Long}_{m0k}}$ and $\sigma^2_{Yr^{Lat}_{m0k}}$ are the between-year variances for the year intercepts for the mean model, $\sigma^2_{ID^{Long}_{sd0j_{exp}}}$ and $\sigma^2_{ID^{Lat}_{sd0j_{exp}}}$ are the between-individual variances for the individual intercepts for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $\sigma^2_{Yr^{Long}_{sd0k_{exp}}}$ and $\sigma^2_{Yr^{Lat}_{sd0k_{exp}}}$ are the between-year variances for the year intercepts for the dispersion model on the $ln\left(\sigma_{\epsilon_{ijk}}\right)$ scale, $\sigma_{ID^{Long}_{m0j}ID^{Long}_{sd0j_{exp}}}$ and $\sigma_{ID^{Lat}_{m0j}ID^{Lat}_{sd0j_{exp}}}$ are the covariances between individual intercepts in the mean and dispersion models, $\sigma_{Yr^{Long}_{m0k}Yr^{Long}_{sd0k_{exp}}}$ and $\sigma_{Yr^{Lat}_{m0k}Yr^{Lat}_{sd0k_{exp}}}$ are the covariances between year intercepts in the mean and dispersion models.

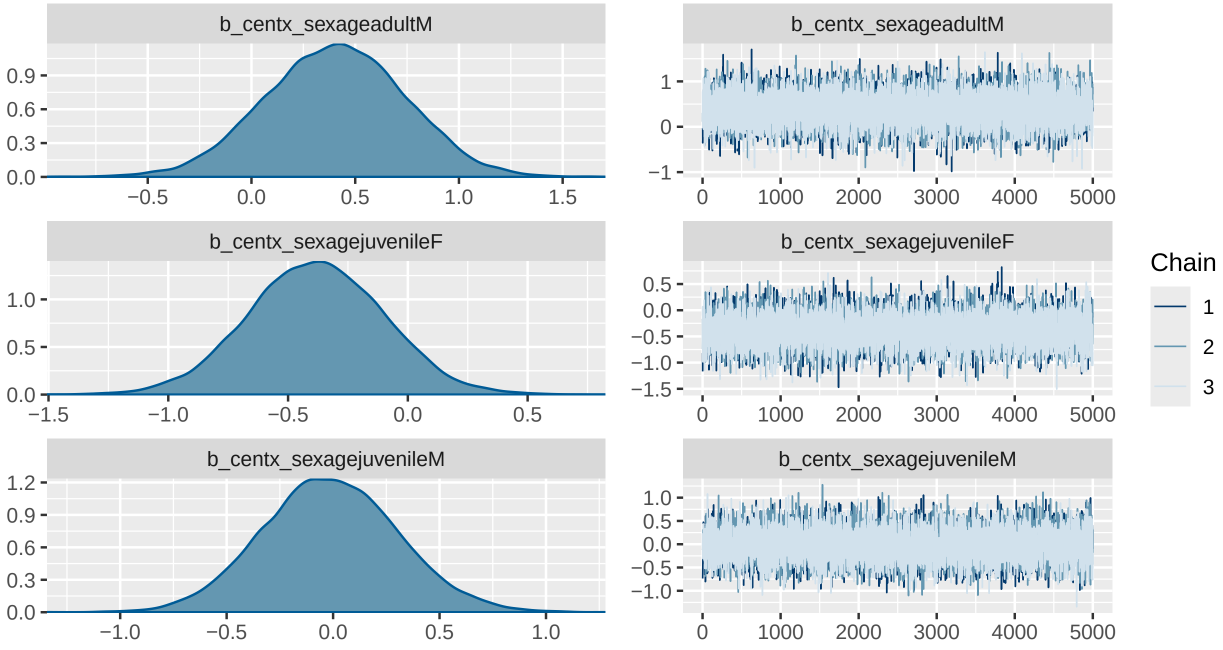
Model code

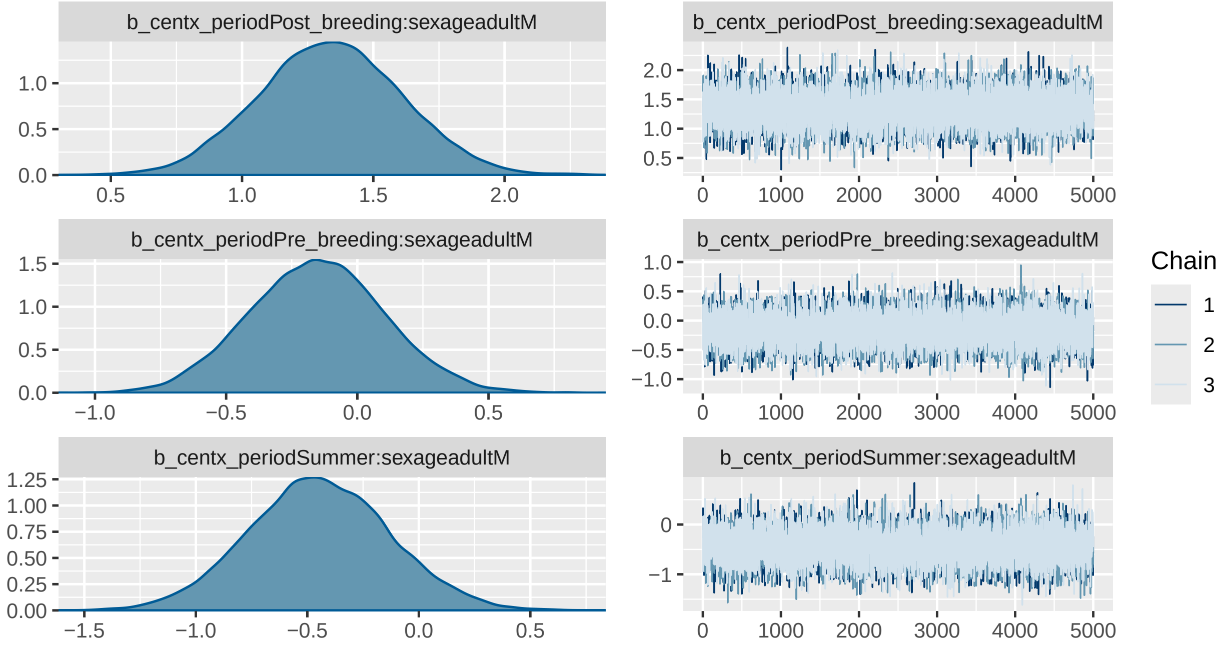
```
d_yr_cent_form <-
  bf(cent.x ~ 0 + Intercept + location + period + sexage + sexage:period +
       s(scale(deploy.yr), by = period) +
       (1|a|id) + (1|b|deploy.yr),
     sigma ~ 0 + Intercept + (1|a|id) + (1|b|deploy.yr),
     autocor = ~ ar(gr = id, p = 1),
     family = brmsfamily("gaussian")) +
  bf(cent.y ~ 0 + Intercept + location + period + sexage + sexage:period +
       s(scale(deploy.yr), by = period) +
       (1|c|id) + (1|d|deploy.yr),
     sigma ~ 0 + Intercept + (1|c|id) + (1|d|deploy.yr),
     autocor = ~ ar(gr = id, p = 1),
     family = brmsfamily("gaussian")) +
  set_rescor(F)

d_yr_cent_prior <- c(

  prior(lkj(1), class = "cor", group = "deploy.yr"),
  prior(lkj(1), class = "cor", group = "id"),

  prior(normal(-62, 5), class = "b", coef = "Intercept", resp = "centx"),
  prior(normal(0.5, 0.5), class = "ar", resp = "centx"),

  prior(normal(0, 10), class = "b", resp = "centx", coef = "locationNorthGSL"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "locationSouthGSL"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPost_breeding"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPost_breeding:sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPost_breeding:sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPost_breeding:sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPre_breeding"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPre_breeding:sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPre_breeding:sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodPre_breeding:sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodSummer"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodSummer:sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodSummer:sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "periodSummer:sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sscaledeploy.yr:periodBreeding_1"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sscaledeploy.yr:periodPost_breeding_1"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sscaledeploy.yr:periodPre_breeding_1"),
  prior(normal(0, 10), class = "b", resp = "centx", coef = "sscaledeploy.yr:periodSummer_1"),

  prior(normal(0, 1), class = "sd", group = "deploy.yr", resp = "centx"),
  prior(normal(0, 1), class = "sd", group = "id", resp = "centx"),

  prior(normal(0, 1), class = "sds", coef = "s(scale(deploy.yr), by = period)", resp = "centx"),

  prior(normal(0, 1), class = "b", coef = "Intercept", resp = "centx", dpar = "sigma"),

  prior(normal(0, 1), class = "sd", group = "deploy.yr", resp = "centx", dpar = "sigma"),
  prior(normal(0, 1), class = "sd", group = "id", resp = "centx", dpar = "sigma"),


  prior(normal(45.5, 5), class = "b", coef = "Intercept", resp = "centy"),
  prior(normal(0.5, 0.5), class = "ar", resp = "centy"),

  prior(normal(0, 10), class = "b", resp = "centy", coef = "locationNorthGSL"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "locationSouthGSL"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPost_breeding"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPost_breeding:sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPost_breeding:sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPost_breeding:sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPre_breeding"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPre_breeding:sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPre_breeding:sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodPre_breeding:sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodSummer"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodSummer:sexageadultM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodSummer:sexagejuvenileF"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "periodSummer:sexagejuvenileM"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sscaledeploy.yr:periodBreeding_1"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sscaledeploy.yr:periodPost_breeding_1"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sscaledeploy.yr:periodPre_breeding_1"),
  prior(normal(0, 10), class = "b", resp = "centy", coef = "sscaledeploy.yr:periodSummer_1"),

  prior(normal(0, 1), class = "sd", group = "deploy.yr", resp = "centy"),
  prior(normal(0, 1), class = "sd", group = "id", resp = "centy"),

  prior(normal(0, 1), class = "sds", coef = "s(scale(deploy.yr), by = period)", resp = "centy"),

  prior(normal(0, 1), class = "b", coef = "Intercept", resp = "centy", dpar = "sigma"),

  prior(normal(0, 1), class = "sd", group = "deploy.yr", resp = "centy", dpar = "sigma"),
  prior(normal(0, 1), class = "sd", group = "id", resp = "centy", dpar = "sigma")
)

d_yr_cent <- brm(
  d_yr_cent_form,
  data = hr.data,
  prior = d_yr_cent_prior,
  sample_prior = "yes",
  save_pars = save_pars(all = TRUE),
  iter = 12000, cores = 3, chains = 3, warmup = 2000, thin = 2,
  control = list(adapt_delta = 0.99),
  file = "Objects/models/centroid.rds",
  file_refit = "on_change")
```

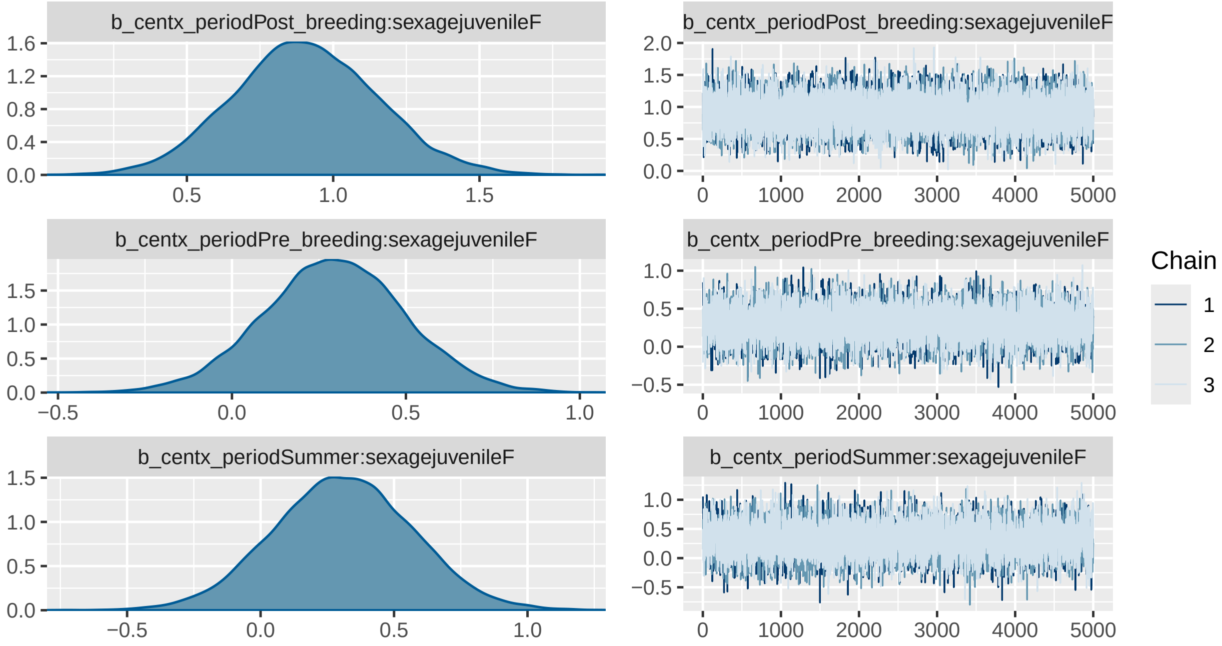
Convergence diagnostic

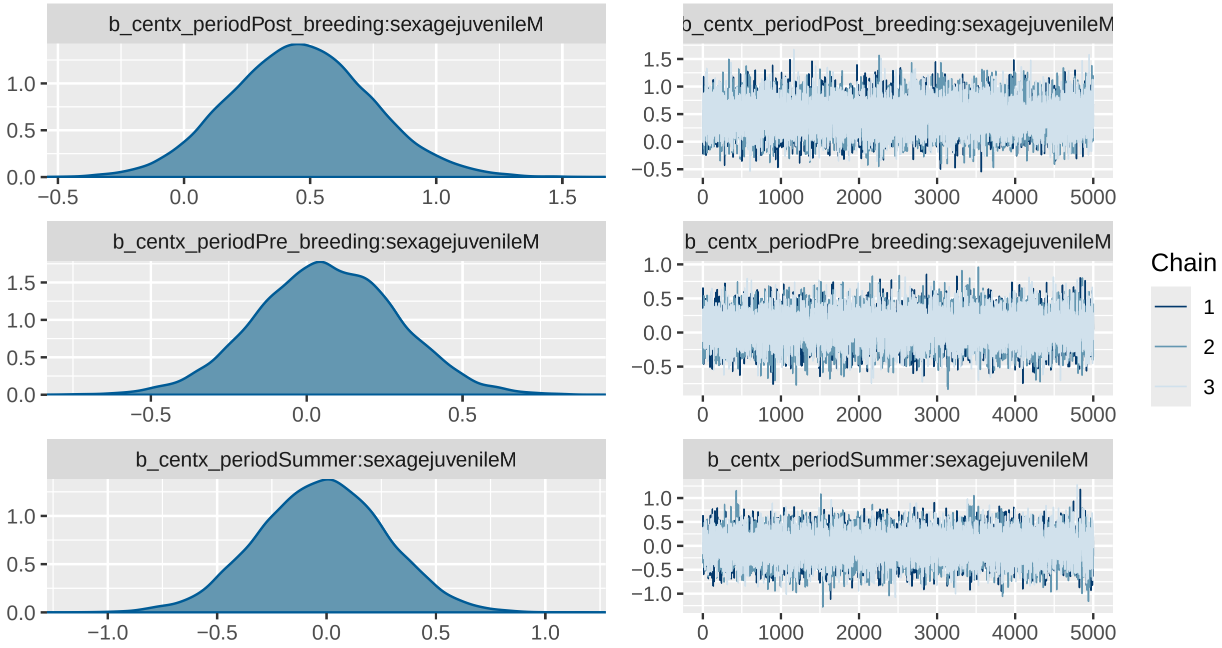



b_centx_periodPost_breeding
2.0
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
b_centx_periodPost_breeding
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
b_centx_periodPre_breeding
2
1
0
−0.75
−0.50
−0.25
0.00
b_centx_periodPre_breeding
0.00
−0.25
−0.50
−0.75
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centx_periodSummer
2.0
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
b_centx_periodSummer
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000

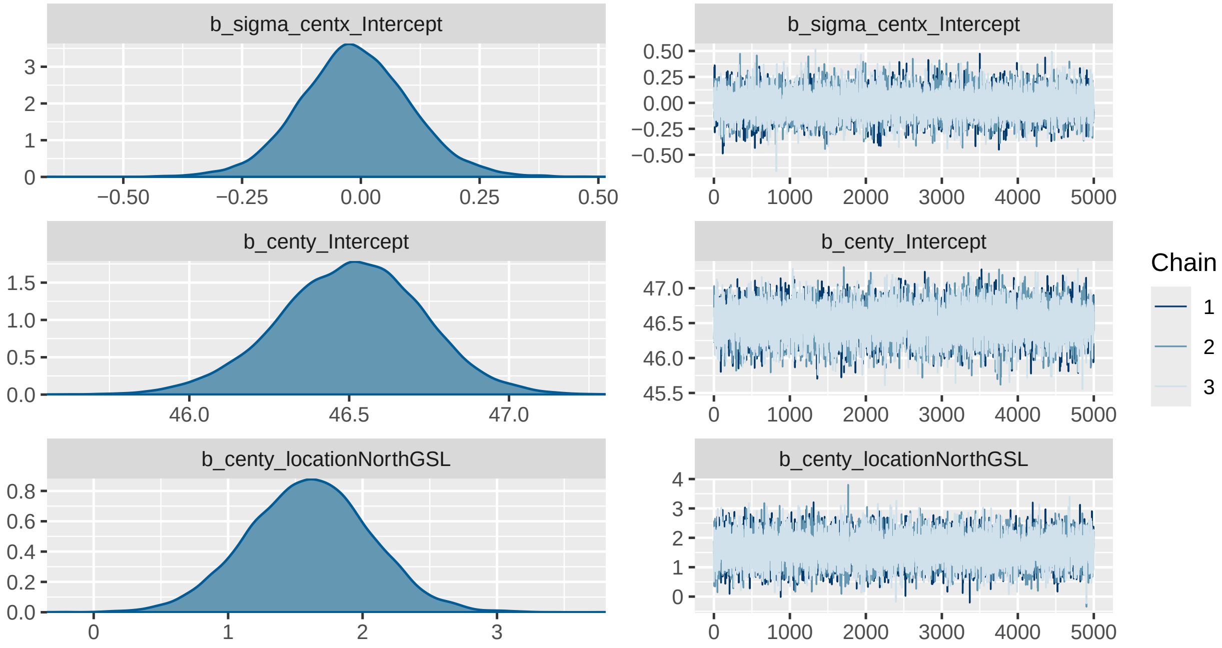

b_centx_sexageadultM
0.9
0.6
0.3
0.0
−0.5
0.0
0.5
1.0
1.5
b_centx_sexageadultM
1
0
−1
0
1000
2000
3000
4000
5000
b_centx_sexagejuvenileF
1.0
0.5
0.0
−1.5
−1.0
−0.5
0.0
0.5
b_centx_sexagejuvenileF
0.5
0.0
−0.5
−1.0
−1.5
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centx_sexagejuvenileM
1.2
0.9
0.6
0.3
0.0
−1.0
−0.5
0.0
0.5
1.0
b_centx_sexagejuvenileM
1.0
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000


b_centx_periodPost_breeding:sexageadultM
b_centx_periodPre_breeding:sexageadultM
b_centx_periodSummer:sexageadultM
0.0
0.5
1.0
1.5
2.0
−1.0
−0.5
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1
0
1000
2000
3000
4000
5000
Chain
1
2
3

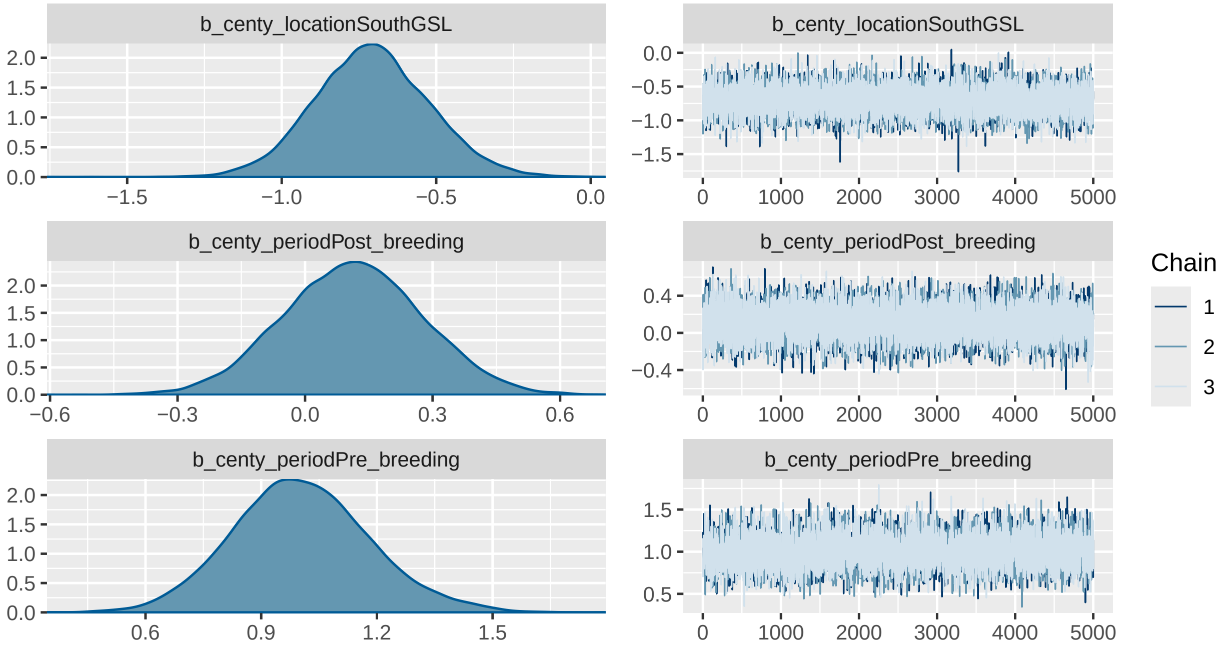

b_centx_periodPost_breeding:sexagejuvenileF
b_centx_periodPre_breeding:sexagejuvenileF
b_centx_periodSummer:sexagejuvenileF
1.6
1.2
0.8
0.4
0.0
0.5
1.0
1.5
2.0
−0.5
0
1000
2000
3000
4000
5000
Chain
1
2
3

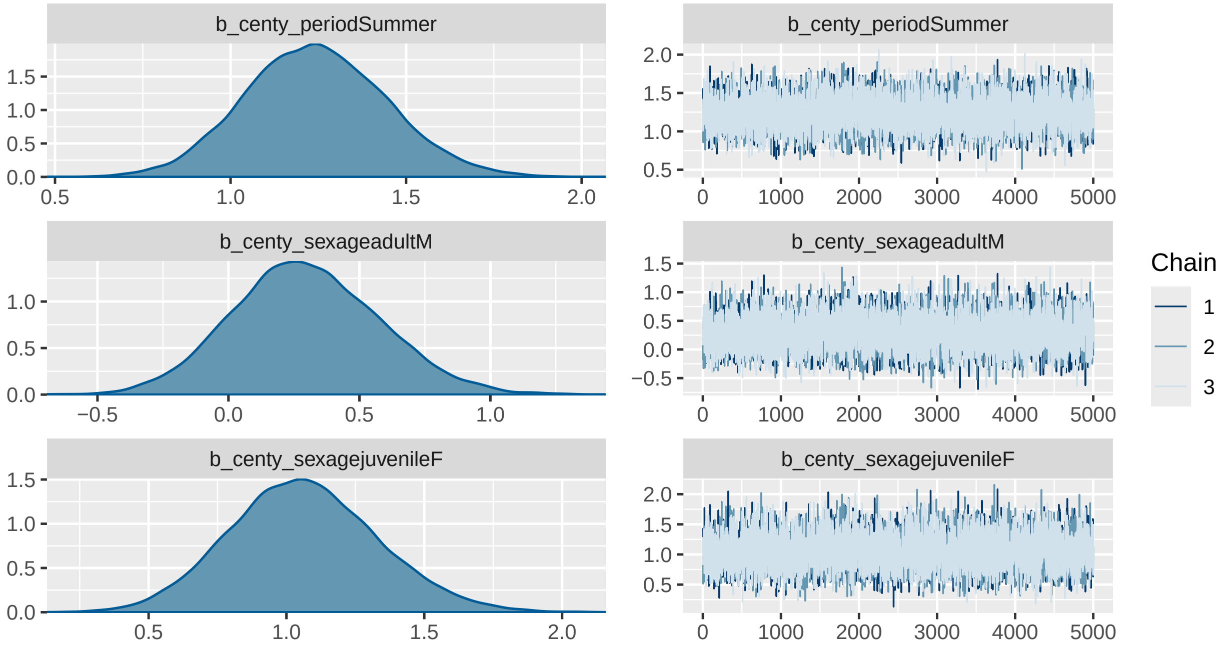


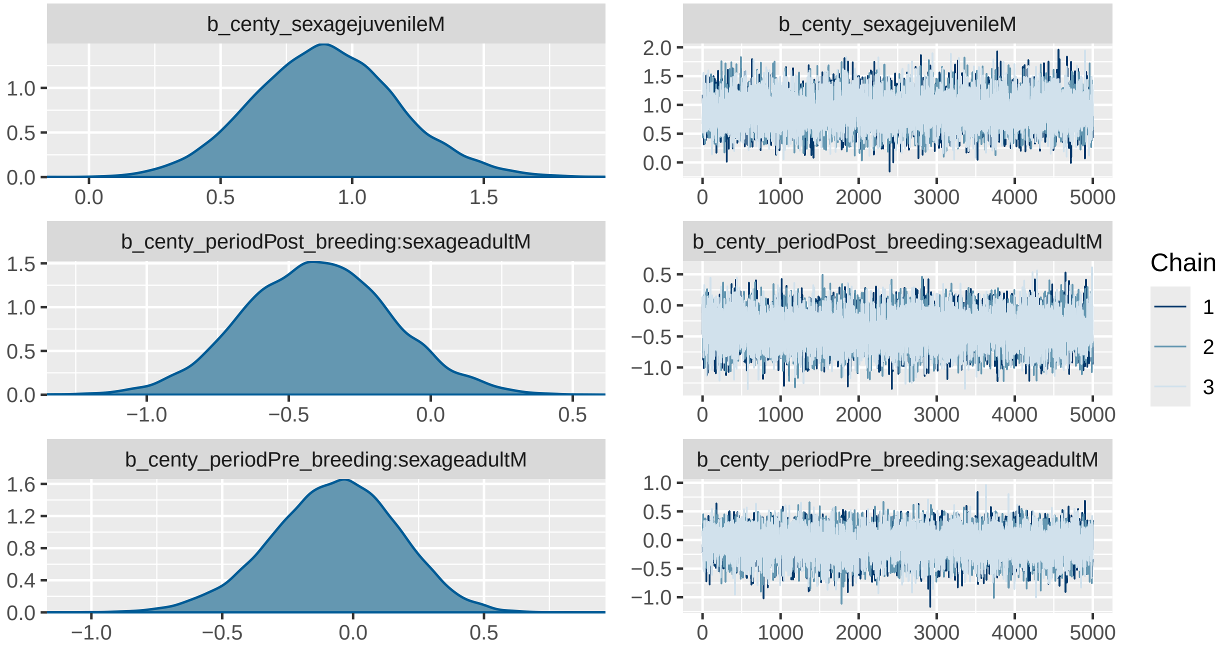
b_centx_periodPost_breeding:sexagejuvenileM
1.0
0.5
0.0
−0.5
0.0
0.5
1.0
1.5
b_centx_periodPost_breeding:sexagejuvenileM
1.5
1.0
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
b_centx_periodPre_breeding:sexagejuvenileM
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
b_centx_periodPre_breeding:sexagejuvenileM
1.0
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centx_periodSummer:sexagejuvenileM
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
b_centx_periodSummer:sexagejuvenileM
1.0
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000

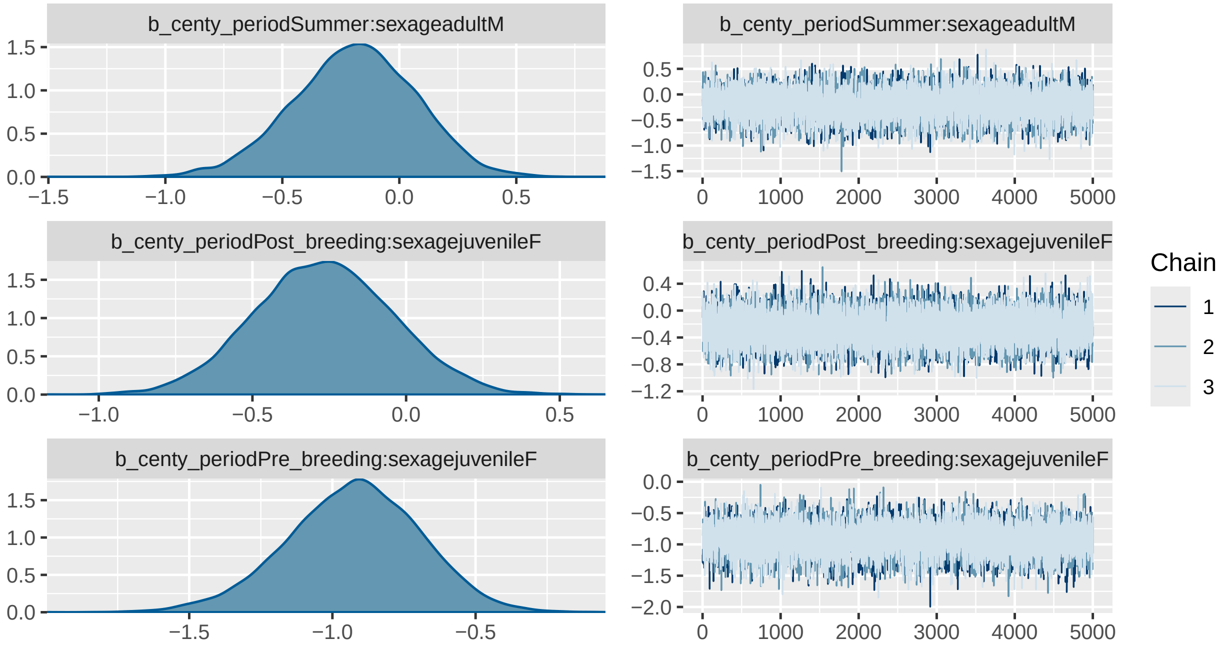


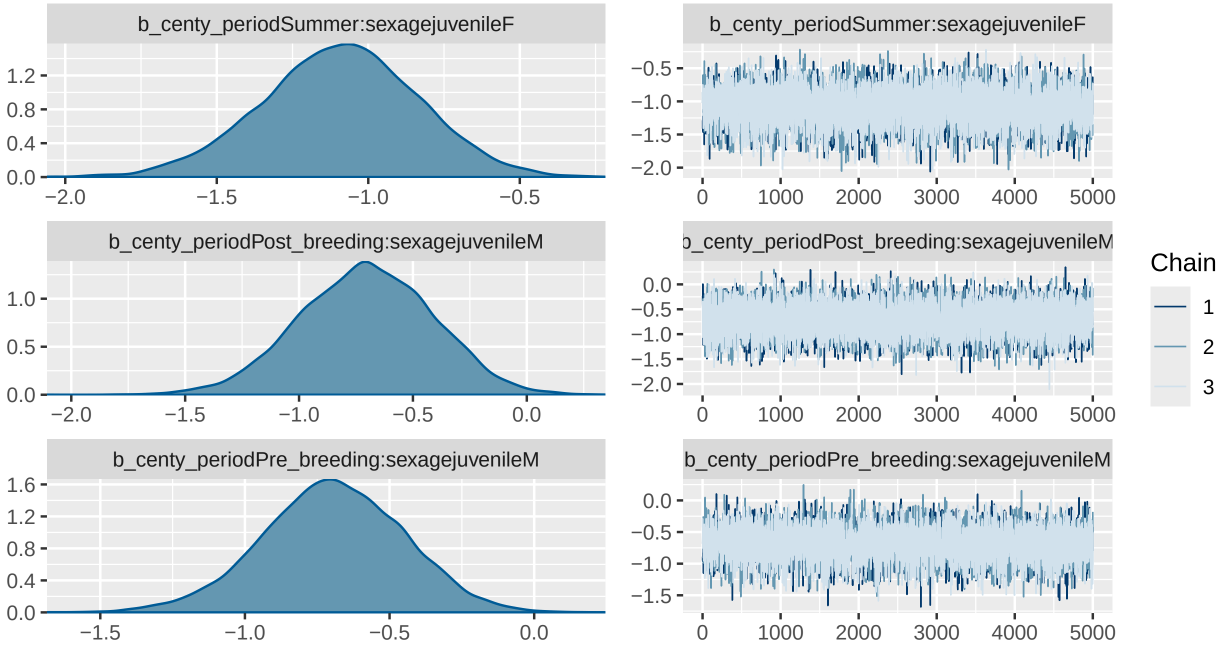
b_sigma_centx_Intercept
3
2
1
0
−0.50
−0.25
0.00
0.25
0.50
b_sigma_centx_Intercept
0.50
0.25
0.00
−0.25
−0.50
0
1000
2000
3000
4000
5000
b_centy_Intercept
1.5
1.0
0.5
0.0
46.0
46.5
47.0
b_centy_Intercept
47.0
46.5
46.0
45.5
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centy_locationNorthGSL
0.8
0.6
0.4
0.2
0.0
0
1
2
3
b_centy_locationNorthGSL
4
3
2
1
0
0
1000
2000
3000
4000
5000

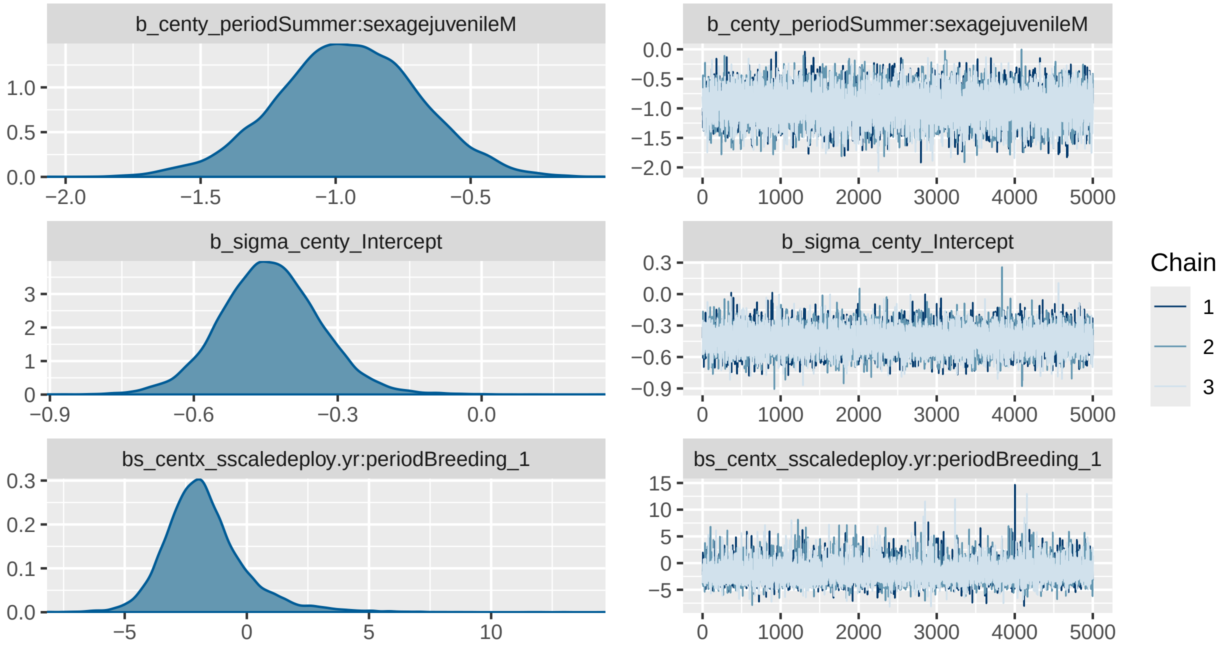

b_centy_locationSouthGSL
2.0
1.5
1.0
0.5
0.0
−1.5
−1.0
−0.5
0.0
b_centy_locationSouthGSL
0.0
−0.5
−1.0
−1.5
0
1000
2000
3000
4000
5000
b_centy_periodPost_breeding
2.0
1.5
1.0
0.5
0.0
−0.6
−0.3
0.0
0.3
0.6
b_centy_periodPost_breeding
0.4
0.0
−0.4
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centy_periodPre_breeding
2.0
1.5
1.0
0.5
0.0
0.6
0.9
1.2
1.5
b_centy_periodPre_breeding
1.5
1.0
0.5
0
1000
2000
3000
4000
5000

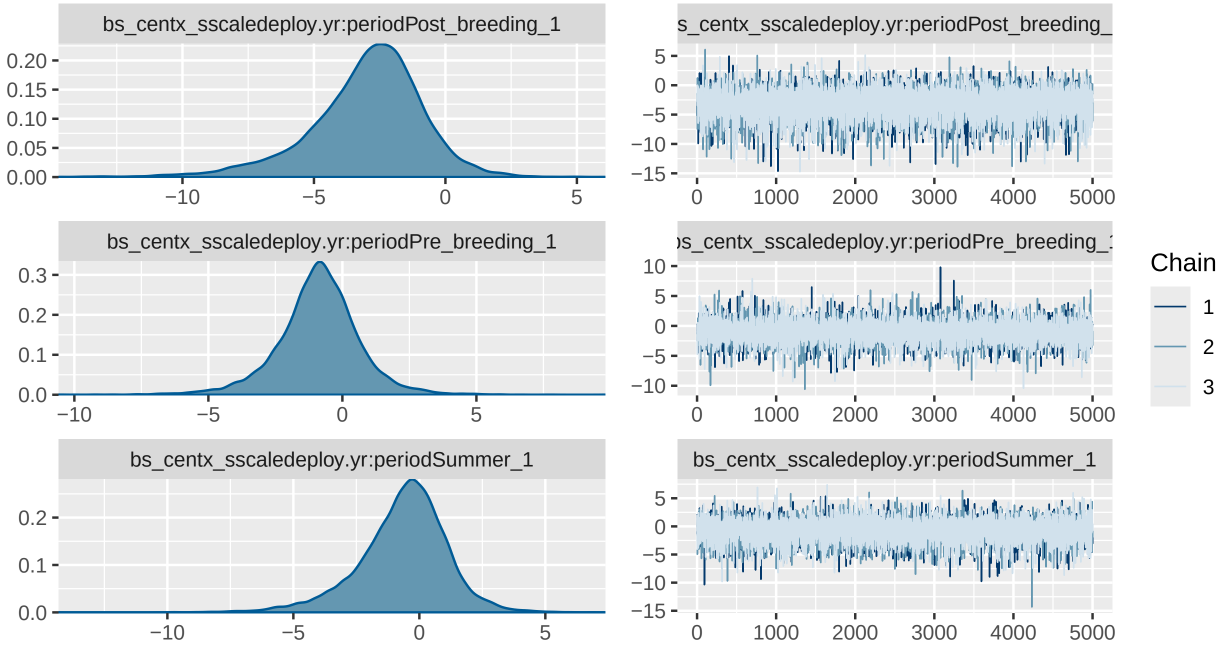

b_centy_periodSummer
1.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0
b_centy_periodSummer
2.0
1.5
1.0
0.5
0
1000
2000
3000
4000
5000
b_centy_sexageadultM
1.0
0.5
0.0
−0.5
0.0
0.5
1.0
b_centy_sexageadultM
1.5
1.0
0.5
0.0
−0.5
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centy_sexagejuvenileF
1.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0
b_centy_sexagejuvenileF
2.0
1.5
1.0
0.5
0
1000
2000
3000
4000
5000


b_centy_sexagejuvenileM
1.0
0.5
0.0
0.0
0.5
1.0
1.5
b_centy_sexagejuvenileM
2.0
1.5
1.0
0.5
0.0
0
1000
2000
3000
4000
5000
b_centy_periodPost_breeding:sexageadultM
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
b_centy_periodPost_breeding:sexageadultM
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centy_periodPre_breeding:sexageadultM
1.6
1.2
0.8
0.4
0.0
−1.0
−0.5
0.0
0.5
b_centy_periodPre_breeding:sexageadultM
1.0
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000

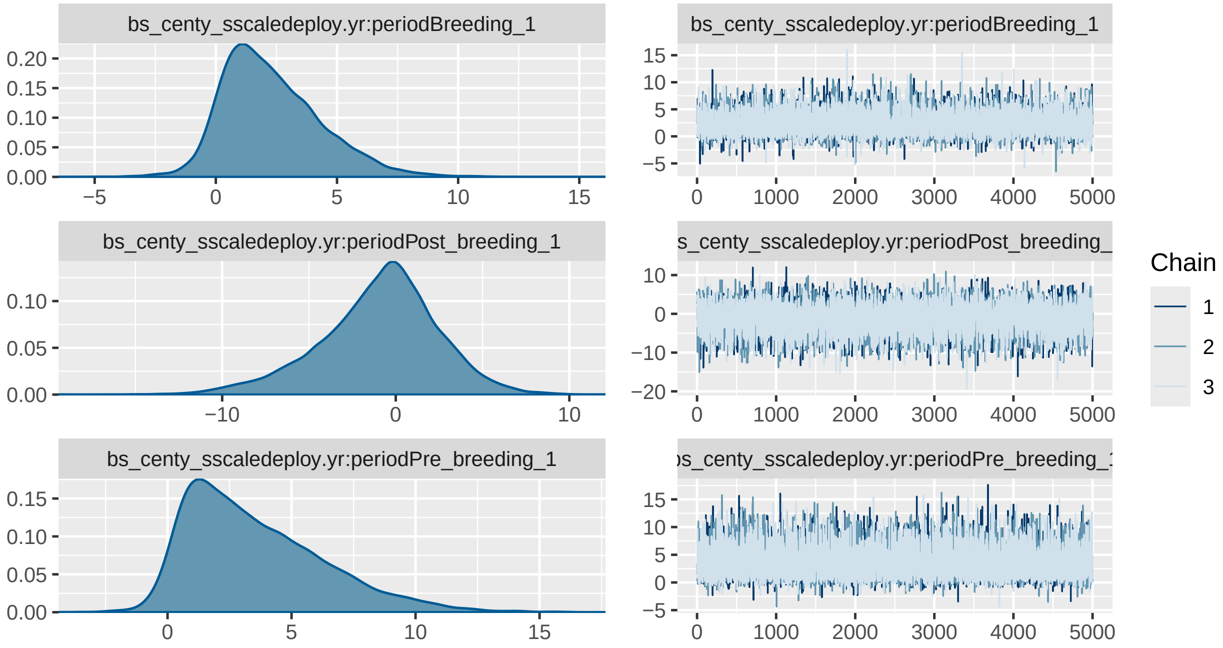

b_centy_periodSummer:sexageadultM
1.5
1.0
0.5
0.0
−1.5
−1.0
−0.5
0.0
0.5
b_centy_periodSummer:sexageadultM
0.5
0.0
−0.5
−1.0
−1.5
0
1000
2000
3000
4000
5000
b_centy_periodPost_breeding:sexagejuvenileF
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
b_centy_periodPost_breeding:sexagejuvenileF
0.4
0.0
−0.4
−0.8
−1.2
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centy_periodPre_breeding:sexagejuvenileF
1.5
1.0
0.5
0.0
−1.5
−1.0
−0.5
b_centy_periodPre_breeding:sexagejuvenileF
0.0
−0.5
−1.0
−1.5
−2.0
0
1000
2000
3000
4000
5000


b_centy_periodSummer:sexagejuvenileF
1.2
0.8
0.4
0.0
−2.0
−1.5
−1.0
−0.5
b_centy_periodSummer:sexagejuvenileF
−0.5
−1.0
−1.5
−2.0
0
1000
2000
3000
4000
5000
b_centy_periodPost_breeding:sexagejuvenileM
1.0
0.5
0.0
−2.0
−1.5
−1.0
−0.5
0.0
b_centy_periodPost_breeding:sexagejuvenileM
0.0
−0.5
−1.0
−1.5
−2.0
0
1000
2000
3000
4000
5000
Chain
1
2
3
b_centy_periodPre_breeding:sexagejuvenileM
1.6
1.2
0.8
0.4
0.0
−1.5
−1.0
−0.5
0.0
b_centy_periodPre_breeding:sexagejuvenileM
0.0
−0.5
−1.0
−1.5
0
1000
2000
3000
4000
5000

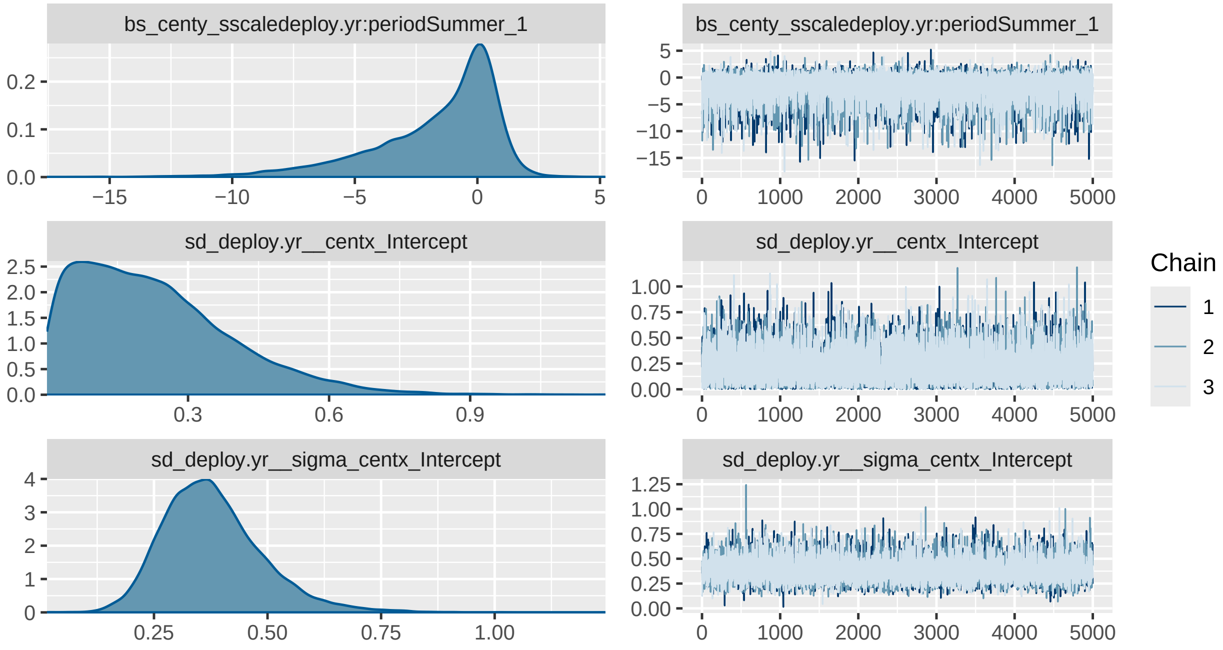

b_centy_periodSummer:sexagejuvenileM
1.0
0.5
0.0
−2.0
−1.5
−1.0
−0.5
b_centy_periodSummer:sexagejuvenileM
0.0
−0.5
−1.0
−1.5
−2.0
0
1000
2000
3000
4000
5000
b_sigma_centy_Intercept
3
2
1
0
−0.9
−0.6
−0.3
0.0
b_sigma_centy_Intercept
0.3
0.0
−0.3
−0.6
−0.9
0
1000
2000
3000
4000
5000
Chain
1
2
3
bs_centx_sscaledeploy.yr:periodBreeding_1
0.3
0.2
0.1
0.0
−5
0
5
10
bs_centx_sscaledeploy.yr:periodBreeding_1
15
10
5
0
−5
0
1000
2000
3000
4000
5000

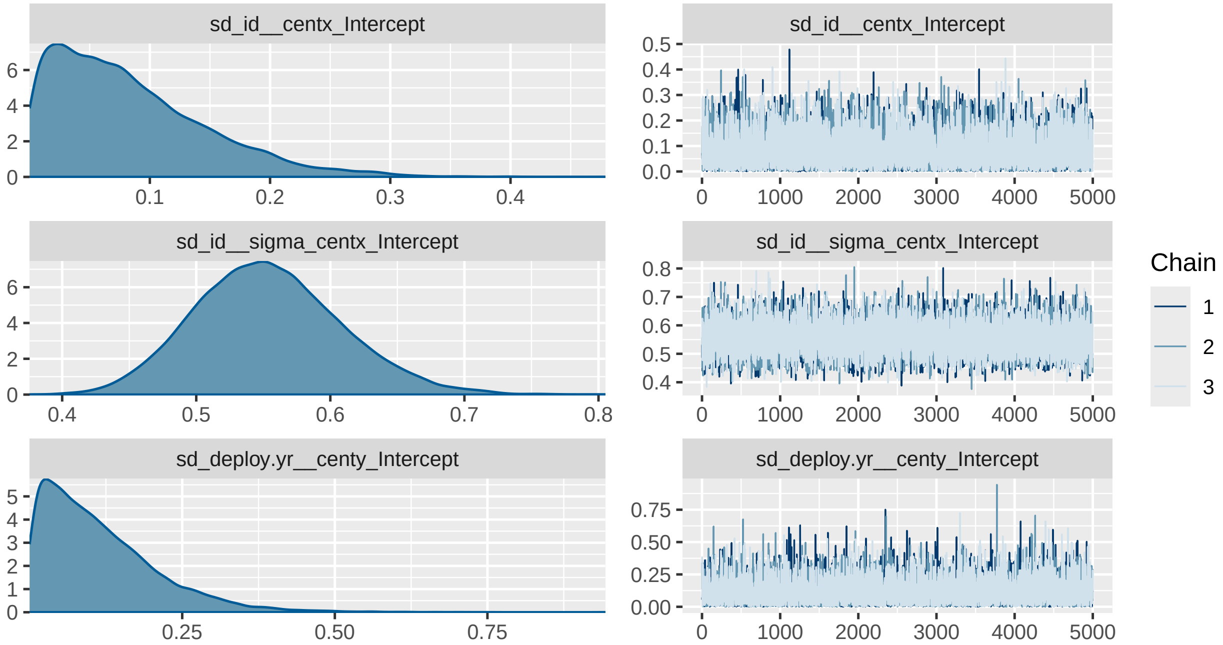


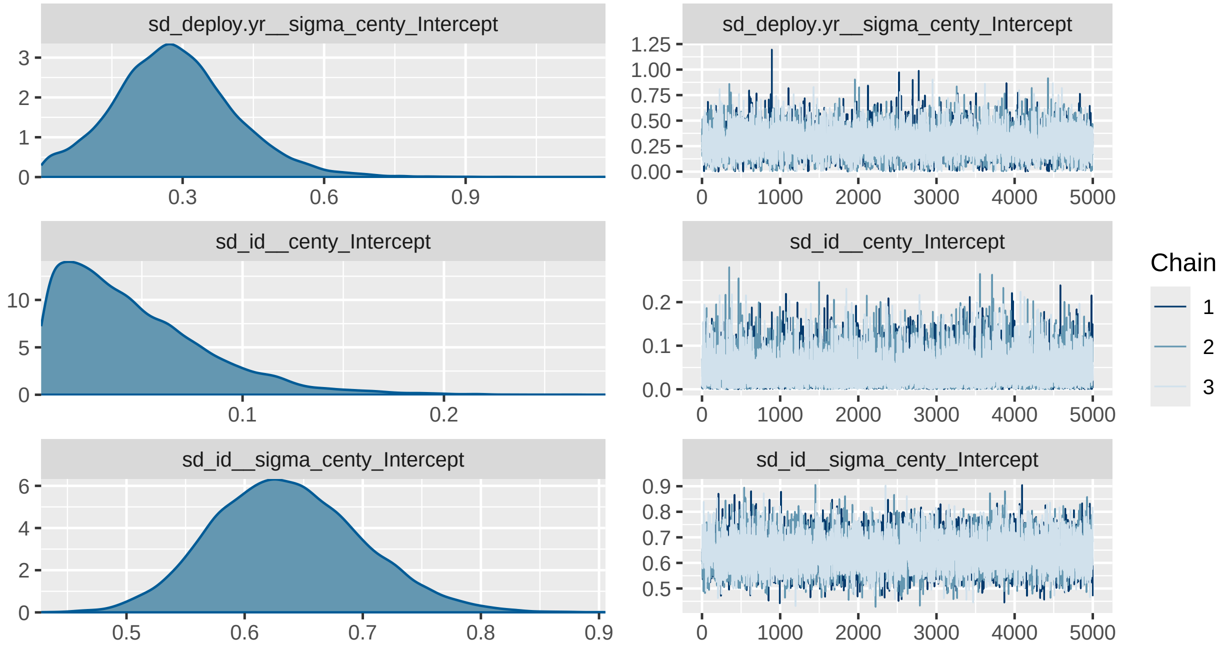
bs_centx_sscaledeploy.yr:periodPost_breeding_1
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
s_centx_sscaledeploy.yr:periodPost_breeding_
5
0
−5
−10
−15
0
1000
2000
3000
4000
5000
bs_centx_sscaledeploy.yr:periodPre_breeding_1
0.3
0.2
0.1
0.0
−10
−5
0
5
bs_centx_sscaledeploy.yr:periodPre_breeding_1
10
5
0
−5
−10
0
1000
2000
3000
4000
5000
Chain
1
2
3
bs_centx_sscaledeploy.yr:periodSummer_1
0.2
0.1
0.0
−10
−5
0
5
bs_centx_sscaledeploy.yr:periodSummer_1
5
0
−5
−10
−15
0
1000
2000
3000
4000
5000

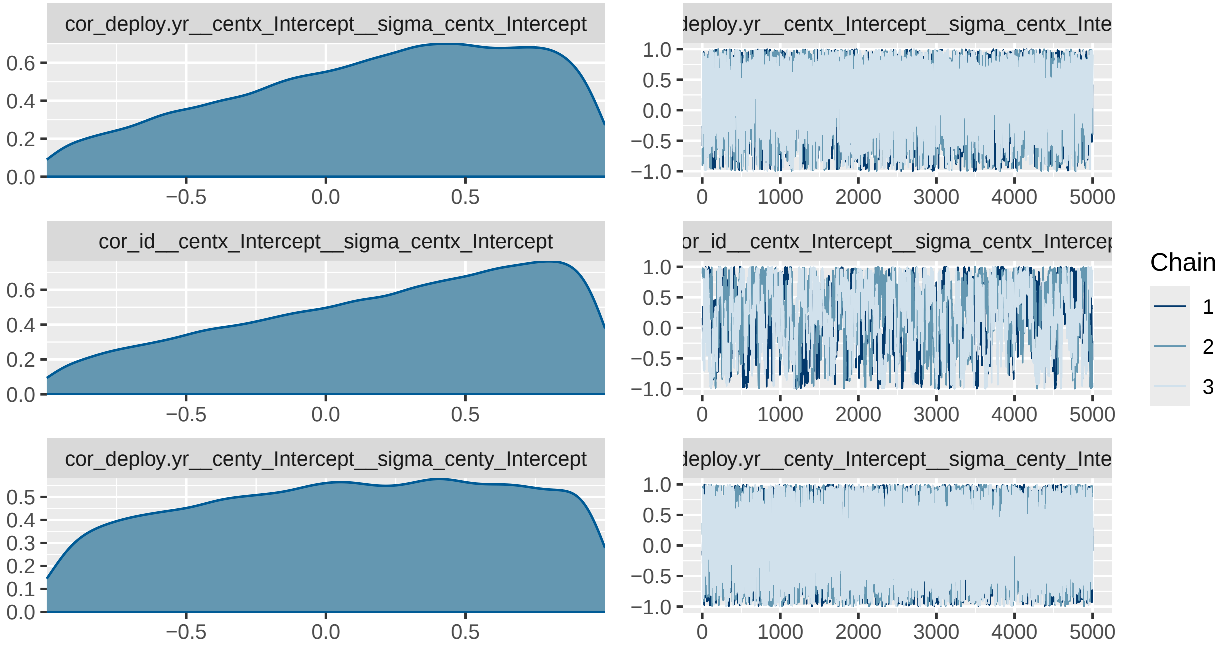


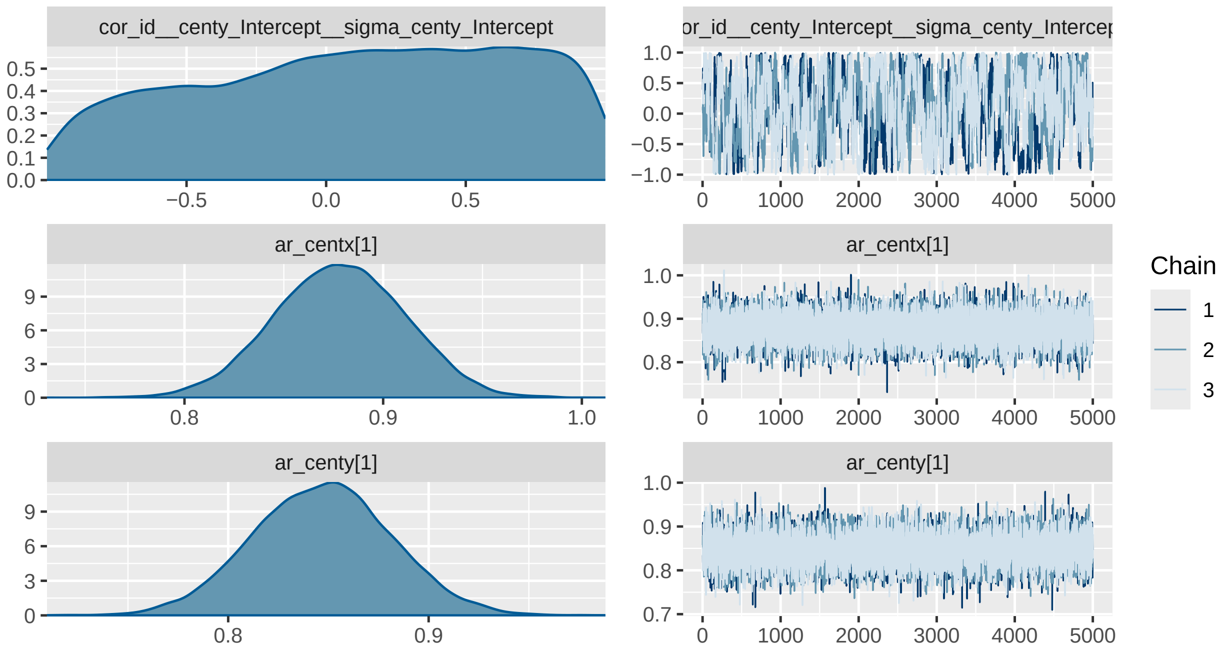
bs_centy_sscaledeploy.yr:periodBreeding_1
0.20
0.15
0.10
0.05
0.00
−5
0
5
10
15
bs_centy_sscaledeploy.yr:periodBreeding_1
15
10
5
0
−5
0
1000
2000
3000
4000
5000
bs_centy_sscaledeploy.yr:periodPost_breeding_1
0.10
0.05
0.00
−10
0
10
s_centy_sscaledeploy.yr:periodPost_breeding_
10
0
−10
−20
0
1000
2000
3000
4000
5000
Chain
1
2
3
bs_centy_sscaledeploy.yr:periodPre_breeding_1
0.15
0.10
0.05
0.00
0
5
10
15
bs_centy_sscaledeploy.yr:periodPre_breeding_1
15
10
5
0
−5
0
1000
2000
3000
4000
5000

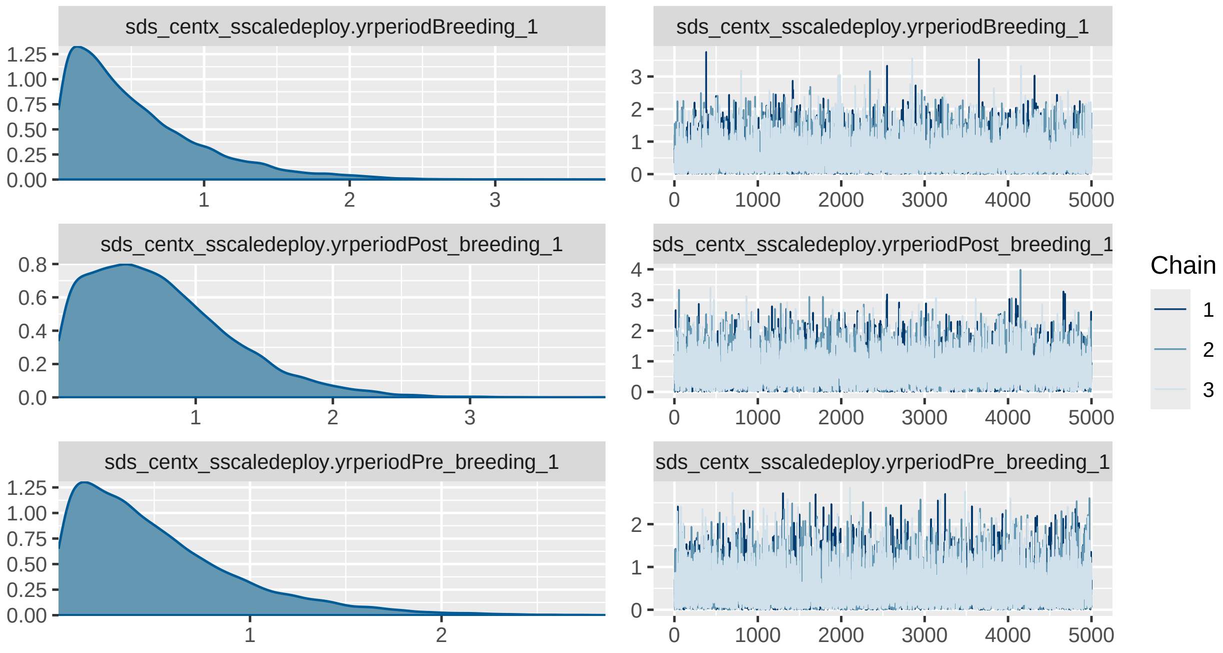

bs_centy_sscaledeploy.yr:periodSummer_1
0.2
0.1
0.0
−15
−10
−5
0
5
bs_centy_sscaledeploy.yr:periodSummer_1
5
0
−5
−10
−15
0
1000
2000
3000
4000
5000
sd_deploy.yr__centx_Intercept
2.5
2.0
1.5
1.0
0.5
0.0
0.3
0.6
0.9
sd_deploy.yr__centx_Intercept
1.00
0.75
0.50
0.25
0.00
0
1000
2000
3000
4000
5000
Chain
1
2
3
sd_deploy.yr__sigma_centx_Intercept
4
3
2
1
0
0.25
0.50
0.75
1.00
sd_deploy.yr__sigma_centx_Intercept
1.25
1.00
0.75
0.50
0.25
0.00
0
1000
2000
3000
4000
5000

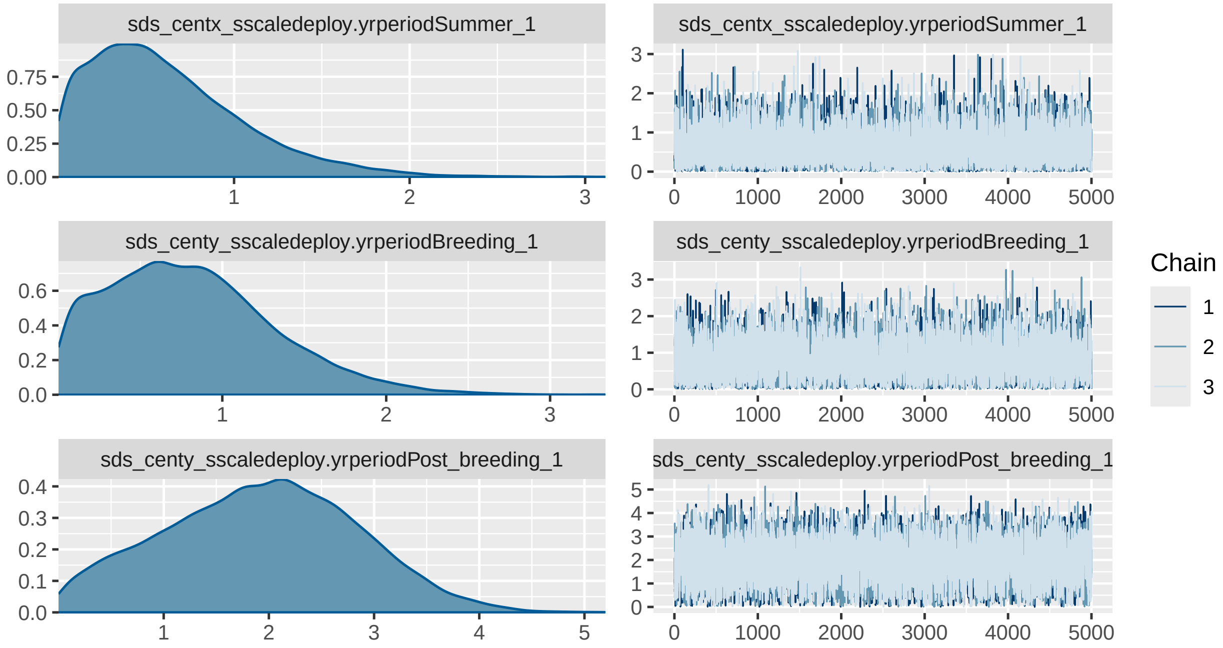

sd_id__centx_Intercept
6
4
2
0
0.1
0.2
0.3
0.4
sd_id__centx_Intercept
0.5
0.4
0.3
0.2
0.1
0.0
0
1000
2000
3000
4000
5000
sd_id__sigma_centx_Intercept
6
4
2
0
0.4
0.5
0.6
0.7
0.8
sd_id__sigma_centx_Intercept
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000
Chain
1
2
3
sd_deploy.yr__centy_Intercept
5
4
3
2
1
0
0.25
0.50
0.75
sd_deploy.yr__centy_Intercept
0.75
0.50
0.25
0.00
0
1000
2000
3000
4000
5000

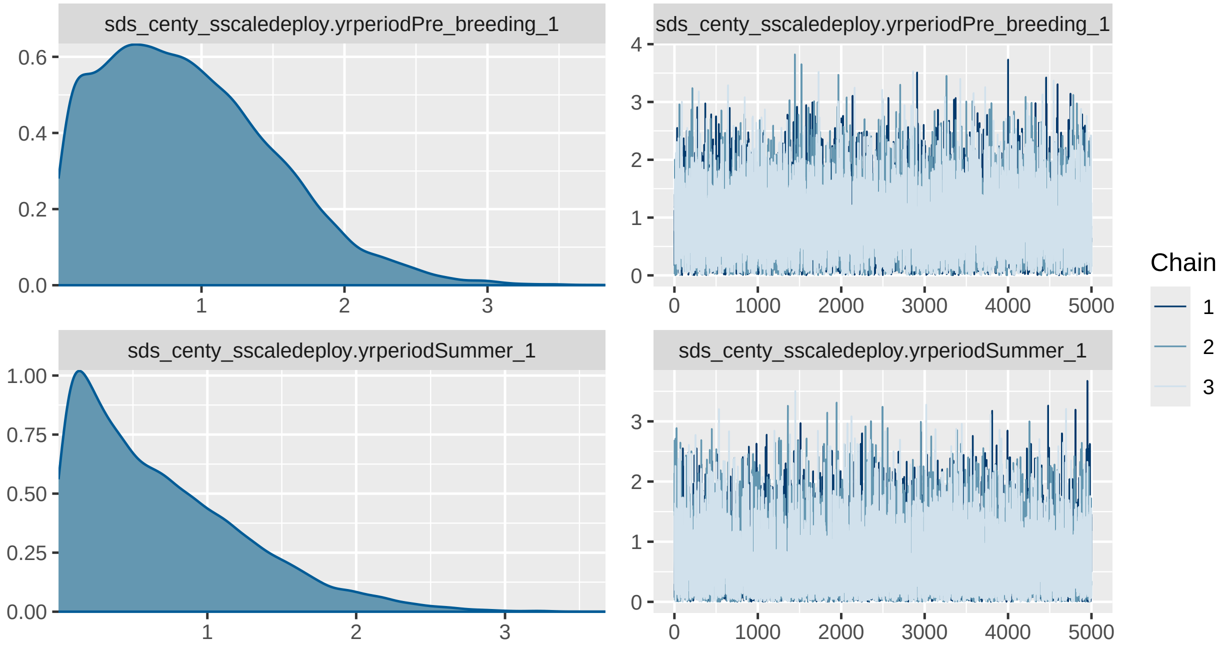

sd_deploy.yr__sigma_centy_Intercept
0
1
2
3
0.3
0.6
0.9
sd_deploy.yr__sigma_centy_Intercept
0.00
0.25
0.50
0.75
1.00
1.25
0
1000
2000
3000
4000
5000
sd_id__centy_Intercept
0
5
10
0.1
0.2
sd_id__centy_Intercept
0.0
0.1
0.2
0
1000
2000
3000
4000
5000
Chain
1
2
3
sd_id__sigma_centy_Intercept
0
2
4
6
0.5
0.6
0.7
0.8
0.9
sd_id__sigma_centy_Intercept
0.5
0.6
0.7
0.8
0.9
0
1000
2000
3000
4000
5000


cor_deploy.yr__centx_Intercept__sigma_centx_Intercept
0.0
0.2
0.4
0.6
−0.5
0.0
0.5
deploy.yr__centx_Intercept__sigma_centx_Inte
−1.0
−0.5
0.0
0.5
1.0
0
1000
2000
3000
4000
5000
cor_id__centx_Intercept__sigma_centx_Intercept
0.0
0.2
0.4
0.6
−0.5
0.0
0.5
or_id__centx_Intercept__sigma_centx_Intercep
−1.0
−0.5
0.0
0.5
1.0
0
1000
2000
3000
4000
5000
Chain
1
2
3
cor_deploy.yr__centy_Intercept__sigma_centy_Intercept
0.0
0.1
0.2
0.3
0.4
0.5
−0.5
0.0
0.5
deploy.yr__centy_Intercept__sigma_centy_Inte
−1.0
−0.5
0.0
0.5
1.0
0
1000
2000
3000
4000
5000

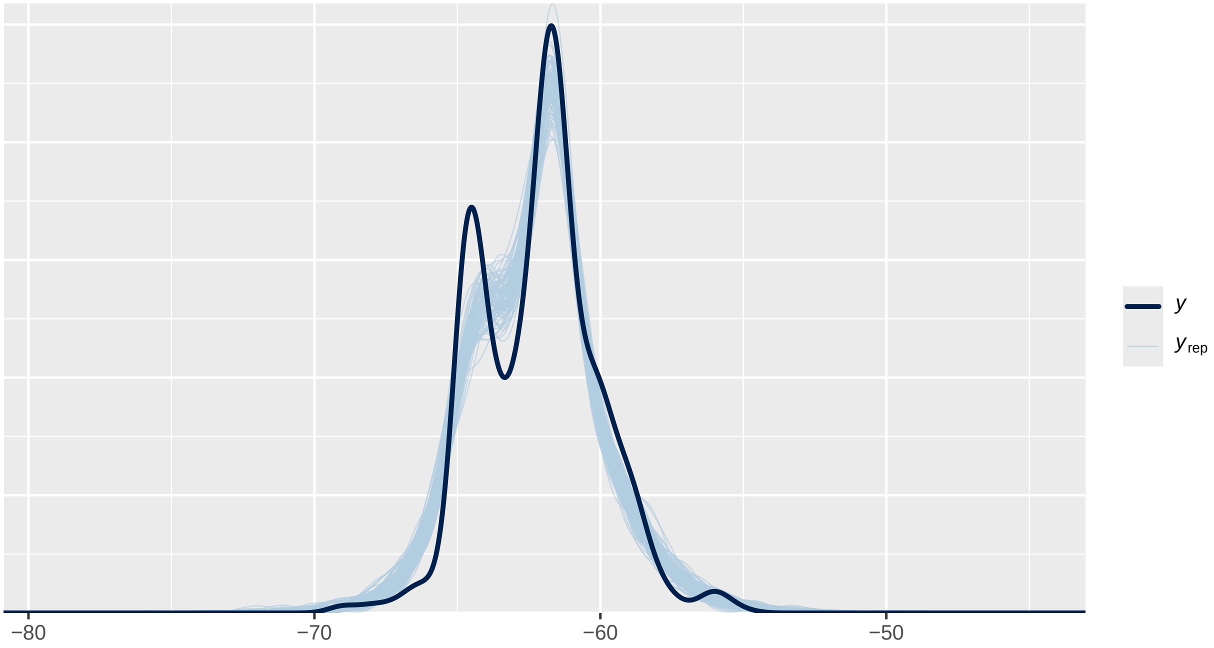

cor_id__centy_Intercept__sigma_centy_Intercept
0.5
0.4
0.3
0.2
0.1
0.0
−0.5
0.0
0.5
or_id__centy_Intercept__sigma_centy_Intercep
1.0
0.5
0.0
−0.5
−1.0
0
1000
2000
3000
4000
5000
ar_centx[1]
9
6
3
0
0.8
0.9
1.0
ar_centx[1]
1.0
0.9
0.8
0
1000
2000
3000
4000
5000
Chain
1
2
3
ar_centy[1]
9
6
3
0
0.8
0.9
ar_centy[1]
1.0
0.9
0.8
0.7
0
1000
2000
3000
4000
5000

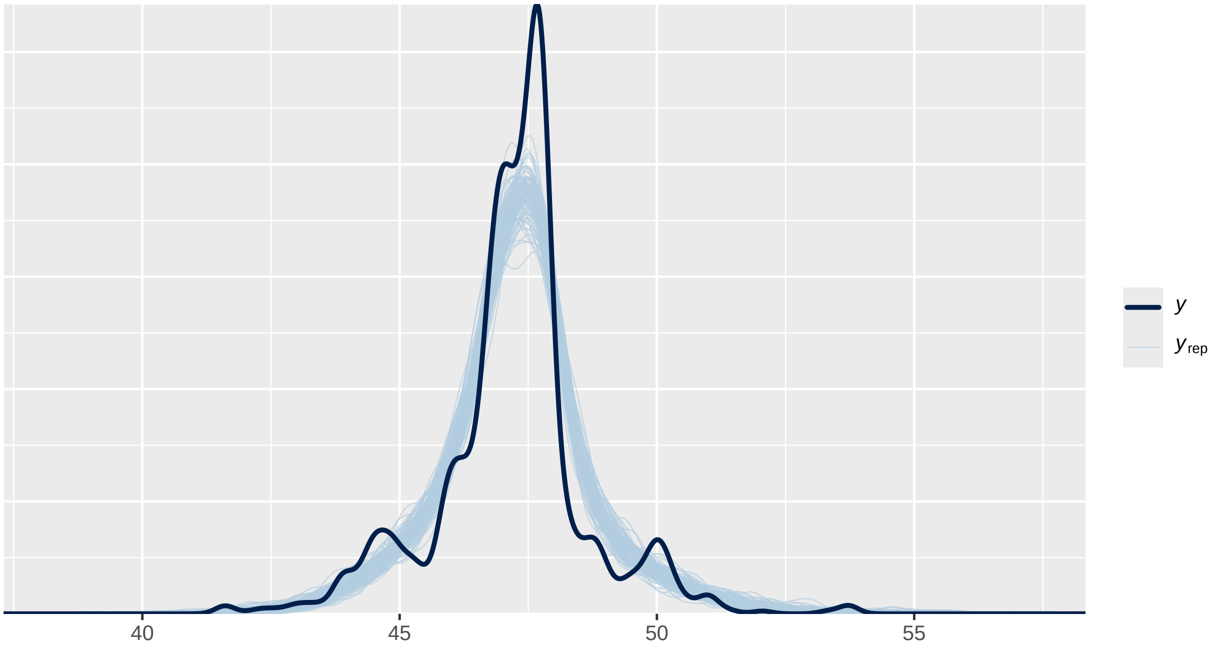

sds_centx_sscaledeploy.yrperiodBreeding_1
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
sds_centx_sscaledeploy.yrperiodBreeding_1
3
2
1
0
0
1000
2000
3000
4000
5000
sds_centx_sscaledeploy.yrperiodPost_breeding_1
0.8
0.6
0.4
0.2
0.0
1
2
3
sds_centx_sscaledeploy.yrperiodPost_breeding_1
4
3
2
1
0
0
1000
2000
3000
4000
5000
Chain
1
2
3
sds_centx_sscaledeploy.yrperiodPre_breeding_1
1.25
1.00
0.75
0.50
0.25
0.00
1
2
sds_centx_sscaledeploy.yrperiodPre_breeding_1
2
1
0
0
1000
2000
3000
4000
5000


sds_centx_sscaledeploy.yrperiodSummer_1
0.75
0.50
0.25
0.00
1
2
3
sds_centx_sscaledeploy.yrperiodSummer_1
3
2
1
0
0
1000
2000
3000
4000
5000
sds_centy_sscaledeploy.yrperiodBreeding_1
0.6
0.4
0.2
0.0
1
2
3
sds_centy_sscaledeploy.yrperiodBreeding_1
3
2
1
0
0
1000
2000
3000
4000
5000
Chain
1
2
3
sds_centy_sscaledeploy.yrperiodPost_breeding_1
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
sds_centy_sscaledeploy.yrperiodPost_breeding_1
5
4
3
2
1
0
0
1000
2000
3000
4000
5000


sds_centy_sscaledeploy.yrperiodPre_breeding_1
0.6
0.4
0.2
0.0
1
2
3
sds_centy_sscaledeploy.yrperiodPre_breeding_1
4
3
2
1
0
0
1000
2000
3000
4000
5000
Chain
1
2
3
sds_centy_sscaledeploy.yrperiodSummer_1
1.00
0.75
0.50
0.25
0.00
1
2
3
sds_centy_sscaledeploy.yrperiodSummer_1
3
2
1
0
0
1000
2000
3000
4000
5000

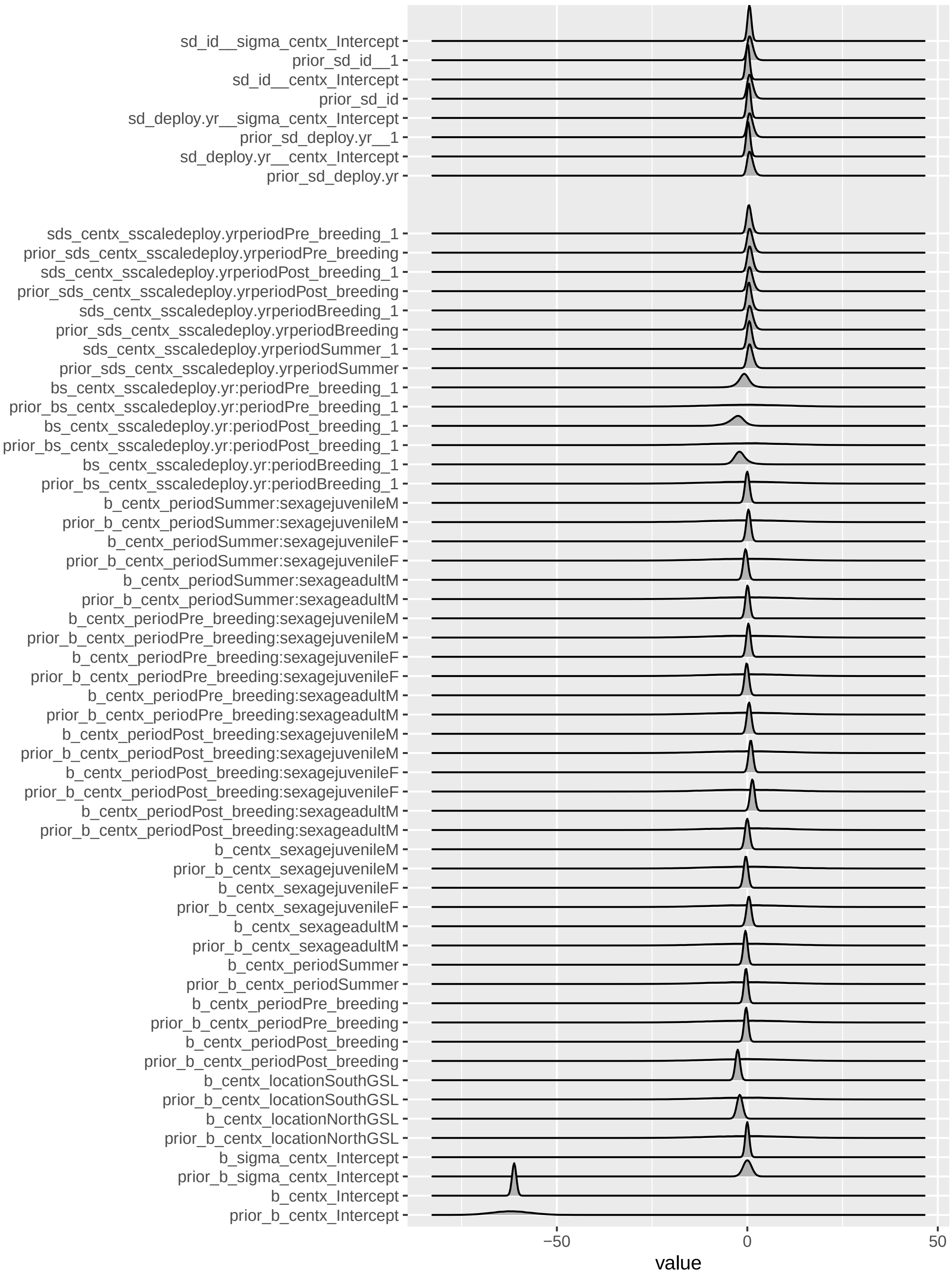
Posterior predictive checks



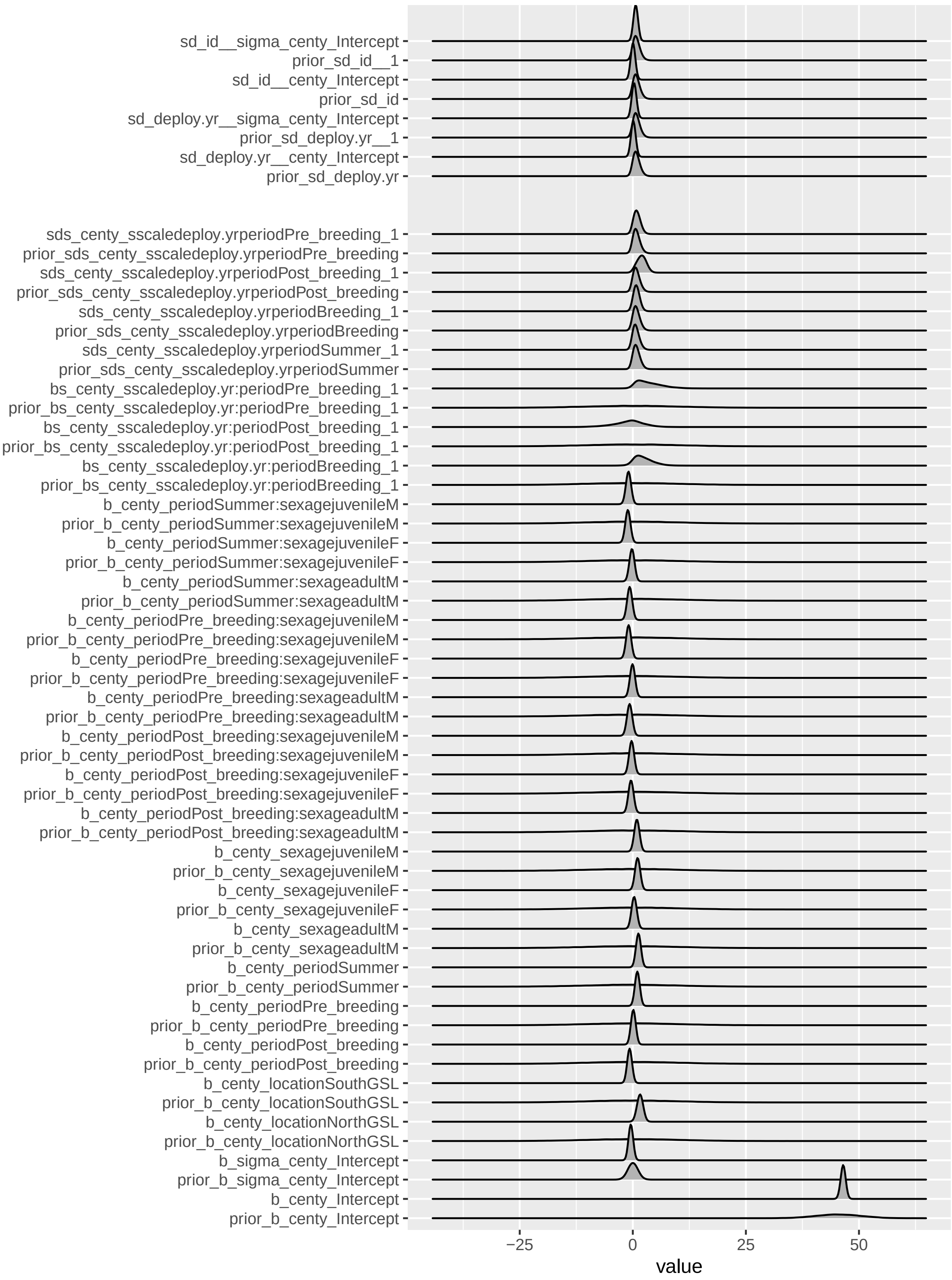


Priors vs posteriors

```
Picking joint bandwidth of 0.487
```


sd_id__sigma_centx_Intercept
prior_sd_id__1
sd_id__centx_Intercept
prior_sd_id
sd_deploy.yr__sigma_centx_Intercept
prior_sd_deploy.yr__1
sd_deploy.yr__centx_Intercept
prior_sd_deploy.yr
sds_centx_sscaledeploy.yrperiodPre_breeding_1
prior_sds_centx_sscaledeploy.yrperiodPre_breeding
sds_centx_sscaledeploy.yrperiodPost_breeding_1
prior_sds_centx_sscaledeploy.yrperiodPost_breeding
sds_centx_sscaledeploy.yrperiodBreeding_1
prior_sds_centx_sscaledeploy.yrperiodBreeding
sds_centx_sscaledeploy.yrperiodSummer_1
prior_sds_centx_sscaledeploy.yrperiodSummer
bs_centx_sscaledeploy.yr:periodPre_breeding_1
prior_bs_centx_sscaledeploy.yr:periodPre_breeding_1
bs_centx_sscaledeploy.yr:periodPost_breeding_1
prior_bs_centx_sscaledeploy.yr:periodPost_breeding_1
bs_centx_sscaledeploy.yr:periodBreeding_1
prior_bs_centx_sscaledeploy.yr:periodBreeding_1
b_centx_periodSummer:sexagejuvenileM
prior_b_centx_periodSummer:sexagejuvenileM
b_centx_periodSummer:sexagejuvenileF
prior_b_centx_periodSummer:sexagejuvenileF
b_centx_periodSummer:sexageadultM
prior_b_centx_periodSummer:sexageadultM
b_centx_periodPre_breeding:sexagejuvenileM
prior_b_centx_periodPre_breeding:sexagejuvenileM
b_centx_periodPre_breeding:sexagejuvenileF
prior_b_centx_periodPre_breeding:sexagejuvenileF
b_centx_periodPre_breeding:sexageadultM
prior_b_centx_periodPre_breeding:sexageadultM
b_centx_periodPost_breeding:sexagejuvenileM
prior_b_centx_periodPost_breeding:sexagejuvenileM
b_centx_periodPost_breeding:sexagejuvenileF
prior_b_centx_periodPost_breeding:sexagejuvenileF
b_centx_periodPost_breeding:sexageadultM
prior_b_centx_periodPost_breeding:sexageadultM
b_centx_sexagejuvenileM
prior_b_centx_sexagejuvenileM
b_centx_sexagejuvenileF
prior_b_centx_sexagejuvenileF
b_centx_sexageadultM
prior_b_centx_sexageadultM
b_centx_periodSummer
prior_b_centx_periodSummer
b_centx_periodPre_breeding
prior_b_centx_periodPre_breeding
b_centx_periodPost_breeding
prior_b_centx_periodPost_breeding
b_centx_locationSouthGSL
prior_b_centx_locationSouthGSL
b_centx_locationNorthGSL
prior_b_centx_locationNorthGSL
b_sigma_centx_Intercept
prior_b_sigma_centx_Intercept
b_centx_Intercept
prior_b_centx_Intercept
−50
0
50
value

```
Picking joint bandwidth of 0.495
```


sd_id__sigma_centy_Intercept
prior_sd_id__1
sd_id__centy_Intercept
prior_sd_id
sd_deploy.yr__sigma_centy_Intercept
prior_sd_deploy.yr__1
sd_deploy.yr__centy_Intercept
prior_sd_deploy.yr
sds_centy_sscaledeploy.yrperiodPre_breeding_1
prior_sds_centy_sscaledeploy.yrperiodPre_breeding
sds_centy_sscaledeploy.yrperiodPost_breeding_1
prior_sds_centy_sscaledeploy.yrperiodPost_breeding
sds_centy_sscaledeploy.yrperiodBreeding_1
prior_sds_centy_sscaledeploy.yrperiodBreeding
sds_centy_sscaledeploy.yrperiodSummer_1
prior_sds_centy_sscaledeploy.yrperiodSummer
bs_centy_sscaledeploy.yr:periodPre_breeding_1
prior_bs_centy_sscaledeploy.yr:periodPre_breeding_1
bs_centy_sscaledeploy.yr:periodPost_breeding_1
prior_bs_centy_sscaledeploy.yr:periodPost_breeding_1
bs_centy_sscaledeploy.yr:periodBreeding_1
prior_bs_centy_sscaledeploy.yr:periodBreeding_1
b_centy_periodSummer:sexagejuvenileM
prior_b_centy_periodSummer:sexagejuvenileM
b_centy_periodSummer:sexagejuvenileF
prior_b_centy_periodSummer:sexagejuvenileF
b_centy_periodSummer:sexageadultM
prior_b_centy_periodSummer:sexageadultM
b_centy_periodPre_breeding:sexagejuvenileM
prior_b_centy_periodPre_breeding:sexagejuvenileM
b_centy_periodPre_breeding:sexagejuvenileF
prior_b_centy_periodPre_breeding:sexagejuvenileF
b_centy_periodPre_breeding:sexageadultM
prior_b_centy_periodPre_breeding:sexageadultM
b_centy_periodPost_breeding:sexagejuvenileM
prior_b_centy_periodPost_breeding:sexagejuvenileM
b_centy_periodPost_breeding:sexagejuvenileF
prior_b_centy_periodPost_breeding:sexagejuvenileF
b_centy_periodPost_breeding:sexageadultM
prior_b_centy_periodPost_breeding:sexageadultM
b_centy_sexagejuvenileM
prior_b_centy_sexagejuvenileM
b_centy_sexagejuvenileF
prior_b_centy_sexagejuvenileF
b_centy_sexageadultM
prior_b_centy_sexageadultM
b_centy_periodSummer
prior_b_centy_periodSummer
b_centy_periodPre_breeding
prior_b_centy_periodPre_breeding
b_centy_periodPost_breeding
prior_b_centy_periodPost_breeding
b_centy_locationSouthGSL
prior_b_centy_locationSouthGSL
b_centy_locationNorthGSL
prior_b_centy_locationNorthGSL
b_sigma_centy_Intercept
prior_b_sigma_centy_Intercept
b_centy_Intercept
prior_b_centy_Intercept
−25
0
25
50
value

Model summary

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Population-Level Effects:							
centx_Intercept	-61.206	0.256	-61.704	-60.702	1.000	7 187	12 012
centx_locationNorthGSL	-1.986	0.570	-3.087	-0.862	1.000	12 489	13 341
centx_locationSouthGSL	-2.528	0.312	-3.136	-1.903	1.000	12 086	12 536
centx_periodPost_breeding	-0.300	0.172	-0.643	0.038	1.000	13 179	13 042
centx_periodPre_breeding	-0.350	0.138	-0.624	-0.081	1.000	9 081	12 516
centx_periodSummer	-0.479	0.189	-0.851	-0.119	1.000	9 214	12 086
centx_sexageadultM	0.403	0.338	-0.258	1.050	1.000	10 617	13 214
centx_sexagejuvenileF	-0.384	0.284	-0.945	0.171	1.000	9 147	12 086
centx_sexagejuvenileM	-0.014	0.318	-0.627	0.628	1.000	9 785	11 728
centx_periodPost_breeding:sexageadultM	1.337	0.273	0.811	1.873	1.000	12 067	13 006
centx_periodPre_breeding:sexageadultM	-0.153	0.257	-0.647	0.355	1.000	10 414	13 432
centx_periodSummer:sexageadultM	-0.447	0.315	-1.060	0.180	1.000	10 655	13 243
centx_periodPost_breeding:sexagejuvenileF	0.905	0.248	0.429	1.402	1.000	12 447	14 168
centx_periodPre_breeding:sexagejuvenileF	0.294	0.205	-0.111	0.698	1.000	9 565	12 572
centx_periodSummer:sexagejuvenileF	0.309	0.263	-0.206	0.822	1.000	9 317	12 425
centx_periodPost_breeding:sexagejuvenileM	0.462	0.282	-0.079	1.029	1.000	12 520	12 773
centx_periodPre_breeding:sexagejuvenileM	0.065	0.225	-0.375	0.501	1.000	9 352	12 225
centx_periodSummer:sexagejuvenileM	-0.021	0.290	-0.590	0.539	1.000	9 558	12 867
sigma_centx_Intercept	-0.012	0.117	-0.240	0.225	1.000	14 215	13 452
centy_Intercept	46.504	0.225	46.044	46.932	1.000	5 966	9 388
centy_locationNorthGSL	1.603	0.455	0.712	2.492	1.000	12 017	13 288
centy_locationSouthGSL	-0.708	0.186	-1.067	-0.333	1.000	11 432	12 926
centy_periodPost_breeding	0.113	0.164	-0.209	0.435	1.000	11 500	13 250
centy_periodPre_breeding	1.004	0.178	0.668	1.374	1.000	5 472	7 965
centy_periodSummer	1.243	0.201	0.862	1.648	1.000	5 289	8 593
centy_sexageadultM	0.296	0.281	-0.232	0.866	1.000	6 133	9 204
centy_sexagejuvenileF	1.063	0.269	0.558	1.612	1.000	5 481	7 728
centy_sexagejuvenileM	0.900	0.275	0.370	1.465	1.000	5 576	8 824
centy_periodPost_breeding:sexageadultM	-0.398	0.259	-0.905	0.122	1.000	10 915	12 874
centy_periodPre_breeding:sexageadultM	-0.071	0.241	-0.569	0.376	1.000	6 148	9 386
centy_periodSummer:sexageadultM	-0.192	0.262	-0.719	0.299	1.000	5 890	9 961
centy_periodPost_breeding:sexagejuvenileF	-0.267	0.227	-0.707	0.182	1.000	11 304	13 870
centy_periodPre_breeding:sexagejuvenileF	-0.926	0.233	-1.405	-0.489	1.000	5 569	8 404
centy_periodSummer:sexagejuvenileF	-1.093	0.258	-1.615	-0.597	1.000	5 176	7 444
centy_periodPost_breeding:sexagejuvenileM	-0.706	0.294	-1.289	-0.145	1.000	10 317	13 122
centy_periodPre_breeding:sexagejuvenileM	-0.696	0.245	-1.193	-0.229	1.000	5 423	8 827
centy_periodSummer:sexagejuvenileM	-0.956	0.264	-1.485	-0.441	1.000	5 478	8 157
sigma_centy_Intercept	-0.440	0.105	-0.643	-0.230	1.000	13 184	13 738
centx_sscaledeploy.yr:periodBreeding_1	-1.722	1.707	-4.492	2.455	1.000	12 179	11 896
centx_sscaledeploy.yr:periodPost_breeding_1	-2.994	2.151	-8.081	0.718	1.000	12 652	13 261
centx_sscaledeploy.yr:periodPre_breeding_1	-0.892	1.536	-4.160	2.150	1.000	13 146	13 155
centx_sscaledeploy.yr:periodSummer_1	-0.653	1.751	-4.706	2.492	1.000	12 793	13 200
centy_sscaledeploy.yr:periodBreeding_1	2.308	2.038	-0.829	6.893	1.000	8 674	11 704

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk__ESS	Tail__ESS
centy__sscaledeploy.yr:periodPost__breeding__1	-0.953	3.416	-8.623	5.308	1.000	12 683	13 127
centy__sscaledeploy.yr:periodPre__breeding__1	3.555	2.776	-0.219	10.104	1.000	7 888	12 764
centy__sscaledeploy.yr:periodSummer__1	-1.667	2.593	-8.337	1.432	1.000	8 381	10 250
Smooth Terms:							
sds(centx__sscaledeploy.yrperiodBreeding__1)	0.530	0.462	0.019	1.753	1.000	11 553	13 107
sds(centx__sscaledeploy.yrperiodPost__breeding__1)	0.747	0.519	0.038	1.988	1.000	11 596	11 880
sds(centx__sscaledeploy.yrperiodPre__breeding__1)	0.516	0.429	0.019	1.631	1.000	11 958	13 275
sds(centx__sscaledeploy.yrperiodSummer__1)	0.623	0.443	0.030	1.695	1.000	11 566	11 104
sds(centy__sscaledeploy.yrperiodBreeding__1)	0.809	0.506	0.049	1.959	1.000	6 994	10 630
sds(centy__sscaledeploy.yrperiodPost__breeding__1)	1.918	0.910	0.215	3.664	1.000	6 845	7 236
sds(centy__sscaledeploy.yrperiodPre__breeding__1)	0.916	0.604	0.047	2.287	1.000	7 900	10 162
sds(centy__sscaledeploy.yrperiodSummer__1)	0.696	0.570	0.023	2.096	1.000	7 812	11 816
Group-Level Effects:							
~**Id (Number of levels: 124)**							
sd(centx__Intercept__id)	0.084	0.064	0.003	0.241	1.000	5 139	11 644
sd(sigma__centx__Intercept__id)	0.554	0.055	0.455	0.668	1.000	8 266	10 955
sd(centy__Intercept__id)	0.046	0.036	0.002	0.136	1.001	7 826	12 253
sd(sigma__centy__Intercept__id)	0.636	0.063	0.521	0.768	1.000	6 840	10 566
cor(centx__Intercept__id,sigma__centx__Intercept__id)	0.217	0.531	-0.864	0.967	1.008	456	1 062
cor(centy__Intercept__id,sigma__centy__Intercept__id)	0.102	0.547	-0.906	0.953	1.010	357	964
~**Year (Number of levels: 16)**							
sd(centx__Intercept__yr)	0.228	0.164	0.010	0.617	1.000	8 014	11 493
sd(sigma__centx__Intercept__yr)	0.380	0.110	0.200	0.634	1.000	10 378	11 921
sd(centy__Intercept__yr)	0.115	0.094	0.004	0.346	1.001	7 770	12 106
sd(sigma__centy__Intercept__yr)	0.284	0.129	0.043	0.561	1.000	6 422	6 146
cor(centx__Intercept__yr,sigma__centx__Intercept__yr)	0.182	0.512	-0.859	0.951	1.001	4 399	6 743
cor(centy__Intercept__yr,sigma__centy__Intercept__yr)	0.076	0.550	-0.911	0.954	1.000	6 208	10 699

Post-hoc tests

```
# 'CI': 95%-CI
# '*': The value tested against lies outside the 95%-CI.
# Evidence ratio is a Bayes factor between the hypothesis and its alternative.
#   Values greater than one indicate that evidence in favor of the point
#   hypothesis has increased after seeing the data.

#####################################################################################################################
# Post-hoc comparison between periods, within sexage classes ########################################################
#####################################################################################################################
### Longitude -------------------------------------------------------------------------------------------------------
## Adult females
# Define hypotheses
v_af <- c("Summer = Breeding" = "centx_periodSummer = 0",
          "Summer =  Pre-breeding" = "centx_periodSummer = centx_periodPre_breeding",
          "Summer = Post-breeding" = "centx_periodSummer = centx_periodPost_breeding",
          "Pre-breeding = Breeding" = "centx_periodPre_breeding = 0",
          "Pre-breeding = Post-breeding" = "centx_periodPre_breeding = centx_periodPost_breeding",
          "Breeding = Post-breeding" = "0 = centx_periodPost_breeding")
# Test hypotheses
h_af <- hypothesis(d_yr_cent, v_af)
# Store results
```

```
h_af_tib <- h_af$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Adult males
# Define hypotheses
v_am <- c("Summer = Breeding" =
            "centx_periodSummer + centx_periodSummer:sexageadultM = 0",
          "Summer =  Pre-breeding" =
            "centx_periodSummer + centx_periodSummer:sexageadultM = centx_periodPre_breeding +
             centx_periodPre_breeding:sexageadultM",
          "Summer = Post-breeding" =
            "centx_periodSummer + centx_periodSummer:sexageadultM = centx_periodPost_breeding +
             centx_periodPost_breeding:sexageadultM",
          "Pre-breeding = Breeding" =
            "centx_periodPre_breeding + centx_periodPre_breeding:sexageadultM = 0",
          "Pre-breeding = Post-breeding" =
            "centx_periodPre_breeding + centx_periodPre_breeding:sexageadultM = centx_periodPost_breeding +
             centx_periodPost_breeding:sexageadultM",
          "Breeding = Post-breeding" =
            "0 = centx_periodPost_breeding + centx_periodPost_breeding:sexageadultM")
# Test hypotheses
h_am <- hypothesis(d_yr_cent, v_am)
# Store results
h_am_tib <- h_am$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile females
# Define hypotheses
v_jf <- c("Summer = Breeding" =
            "centx_periodSummer + centx_periodSummer:sexagejuvenileF = 0",
          "Summer =  Pre-breeding" =
            "centx_periodSummer + centx_periodSummer:sexagejuvenileF = centx_periodPre_breeding +
             centx_periodPre_breeding:sexagejuvenileF",
          "Summer = Post-breeding" =
            "centx_periodSummer + centx_periodSummer:sexagejuvenileF = centx_periodPost_breeding +
             centx_periodPost_breeding:sexagejuvenileF",
          "Pre-breeding = Breeding" =
            "centx_periodPre_breeding + centx_periodPre_breeding:sexagejuvenileF = 0",
          "Pre-breeding = Post-breeding" =
            "centx_periodPre_breeding + centx_periodPre_breeding:sexagejuvenileF = centx_periodPost_breeding +
             centx_periodPost_breeding:sexagejuvenileF",
          "Breeding = Post-breeding" =
            "0 = centx_periodPost_breeding + centx_periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_jf <- hypothesis(d_yr_cent, v_jf)
# Store results
h_jf_tib <- h_jf$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile males
# Define hypotheses
v_jm <- c("Summer = Breeding" =
            "centx_periodSummer + centx_periodSummer:sexagejuvenileM = 0",
          "Summer =  Pre-breeding" =
            "centx_periodSummer + centx_periodSummer:sexagejuvenileM = centx_periodPre_breeding +
             centx_periodPre_breeding:sexagejuvenileM",
          "Summer = Post-breeding" =
            "centx_periodSummer + centx_periodSummer:sexagejuvenileM = centx_periodPost_breeding +
             centx_periodPost_breeding:sexagejuvenileM",
          "Pre-breeding = Breeding" =
            "centx_periodPre_breeding + centx_periodPre_breeding:sexagejuvenileM = 0",
          "Pre-breeding = Post-breeding" =
            "centx_periodPre_breeding + centx_periodPre_breeding:sexagejuvenileM = centx_periodPost_breeding +
             centx_periodPost_breeding:sexagejuvenileM",
          "Breeding = Post-breeding" =
            "0 = centx_periodPost_breeding + centx_periodPost_breeding:sexagejuvenileM")
# Test hypotheses
h_jm <- hypothesis(d_yr_cent, v_jm)
# Store results
h_jm_tib <- h_jm$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_centx_seas <-
  full_join(h_af_tib, h_am_tib, by = "Hypothesis") %>%
  full_join(., h_jf_tib, by = "Hypothesis") %>%
  full_join(., h_jm_tib, by = "Hypothesis")

### Latitude -------------------------------------------------------------------------------------------------------
## Adult females
# Define hypotheses
v_af <- c("Summer = Breeding" = "centy_periodSummer = 0",
          "Summer =  Pre-breeding" = "centy_periodSummer = centy_periodPre_breeding",
          "Summer = Post-breeding" = "centy_periodSummer = centy_periodPost_breeding",
          "Pre-breeding = Breeding" = "centy_periodPre_breeding = 0",
          "Pre-breeding = Post-breeding" = "centy_periodPre_breeding = centy_periodPost_breeding",
          "Breeding = Post-breeding" = "0 = centy_periodPost_breeding")
# Test hypotheses
h_af <- hypothesis(d_yr_cent, v_af)
# Store results
h_af_tib <- h_af$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Adult males
# Define hypotheses
v_am <- c("Summer = Breeding" =
            "centy_periodSummer + centy_periodSummer:sexageadultM = 0",
          "Summer =  Pre-breeding" =
            "centy_periodSummer + centy_periodSummer:sexageadultM = centy_periodPre_breeding +
             centy_periodPre_breeding:sexageadultM",
          "Summer = Post-breeding" =
            "centy_periodSummer + centy_periodSummer:sexageadultM = centy_periodPost_breeding +
             centy_periodPost_breeding:sexageadultM",
          "Pre-breeding = Breeding" =
            "centy_periodPre_breeding + centy_periodPre_breeding:sexageadultM = 0",
          "Pre-breeding = Post-breeding" =
            "centy_periodPre_breeding + centy_periodPre_breeding:sexageadultM = centy_periodPost_breeding +
             centy_periodPost_breeding:sexageadultM",
          "Breeding = Post-breeding" =
            "0 = centy_periodPost_breeding + centy_periodPost_breeding:sexageadultM")
# Test hypotheses
h_am <- hypothesis(d_yr_cent, v_am)
# Store results
h_am_tib <- h_am$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile females
# Define hypotheses
v_jf <- c("Summer = Breeding" =
            "centy_periodSummer + centy_periodSummer:sexagejuvenileF = 0",
          "Summer =  Pre-breeding" =
            "centy_periodSummer + centy_periodSummer:sexagejuvenileF = centy_periodPre_breeding +
             centy_periodPre_breeding:sexagejuvenileF",
          "Summer = Post-breeding" =
            "centy_periodSummer + centy_periodSummer:sexagejuvenileF = centy_periodPost_breeding +
             centy_periodPost_breeding:sexagejuvenileF",
          "Pre-breeding = Breeding" =
            "centy_periodPre_breeding + centy_periodPre_breeding:sexagejuvenileF = 0",
          "Pre-breeding = Post-breeding" =
            "centy_periodPre_breeding + centy_periodPre_breeding:sexagejuvenileF = centy_periodPost_breeding +
             centy_periodPost_breeding:sexagejuvenileF",
          "Breeding = Post-breeding" =
            "0 = centy_periodPost_breeding + centy_periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_jf <- hypothesis(d_yr_cent, v_jf)
# Store results
h_jf_tib <- h_jf$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Juvenile males
# Define hypotheses
v_jm <- c("Summer = Breeding" =
            "centy_periodSummer + centy_periodSummer:sexagejuvenileM = 0",
          "Summer =  Pre-breeding" =
            "centy_periodSummer + centy_periodSummer:sexagejuvenileM = centy_periodPre_breeding +
             centy_periodPre_breeding:sexagejuvenileM",
          "Summer = Post-breeding" =
            "centy_periodSummer + centy_periodSummer:sexagejuvenileM = centy_periodPost_breeding +
             centy_periodPost_breeding:sexagejuvenileM",
          "Pre-breeding = Breeding" =
            "centy_periodPre_breeding + centy_periodPre_breeding:sexagejuvenileM = 0",
          "Pre-breeding = Post-breeding" =
            "centy_periodPre_breeding + centy_periodPre_breeding:sexagejuvenileM = centy_periodPost_breeding +
             centy_periodPost_breeding:sexagejuvenileM",
          "Breeding = Post-breeding" =
            "0 = centy_periodPost_breeding + centy_periodPost_breeding:sexagejuvenileM")
# Test hypotheses
h_jm <- hypothesis(d_yr_cent, v_jm)
# Store results
h_jm_tib <- h_jm$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_centy_seas <-
  full_join(h_af_tib, h_am_tib, by = "Hypothesis") %>%
  full_join(., h_jf_tib, by = "Hypothesis") %>%
  full_join(., h_jm_tib, by = "Hypothesis")

####################################################################################################################
# Post-hoc comparison between sexage classes, within periods #######################################################
####################################################################################################################
### Longitude ------------------------------------------------------------------------------------------------------
## Summer
# Define hypotheses
```

```
v_sum <- c("Adult F = Adult M" =
             "0 = centx_sexageadultM + centx_periodSummer:sexageadultM",
          "Adult F = Juvenile F" =
            "0 = centx_sexagejuvenileF + centx_periodSummer:sexagejuvenileF",
          "Adult F = Juvenile M" =
            "0 = centx_sexagejuvenileM + centx_periodSummer:sexagejuvenileM",
          "Adult M = Juvenile M" =
            "centx_sexageadultM + centx_periodSummer:sexageadultM = centx_sexagejuvenileM +
             centx_periodSummer:sexagejuvenileM",
          "Adult M = Juvenile F" =
            "centx_sexageadultM + centx_periodSummer:sexageadultM = centx_sexagejuvenileF +
             centx_periodSummer:sexagejuvenileF",
          "Juvenile M = Juvenile F" =
            "centx_sexagejuvenileM + centx_periodSummer:sexagejuvenileM = centx_sexagejuvenileF +
             centx_periodSummer:sexagejuvenileF")
# Test hypotheses
h_sum <- hypothesis(d_yr_cent, v_sum)
# Store results
h_sum_tib <- h_sum$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Pre-breeding
# Define hypotheses
v_pre <- c("Adult F = Adult M" =
             "0 = centx_sexageadultM + centx_periodPre_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = centx_sexagejuvenileF + centx_periodPre_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = centx_sexagejuvenileM + centx_periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "centx_sexageadultM + centx_periodPre_breeding:sexageadultM = centx_sexagejuvenileM +
             centx_periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "centx_sexageadultM + centx_periodPre_breeding:sexageadultM = centx_sexagejuvenileF +
             centx_periodPre_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "centx_sexagejuvenileM + centx_periodPre_breeding:sexagejuvenileM = centx_sexagejuvenileF +
              centx_periodPre_breeding:sexagejuvenileF")
# Test hypotheses
h_pre <- hypothesis(d_yr_cent, v_pre)
# Store results
h_pre_tib <- h_pre$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Breeding
# Define hypotheses
v_bred <- c("Adult F = Adult M" = "0 = centx_sexageadultM ",
           "Adult F = Juvenile F" = "0 = centx_sexagejuvenileF",
           "Adult F = Juvenile M" = "0 = centx_sexagejuvenileM",
           "Adult M = Juvenile M" = "centx_sexageadultM = centx_sexagejuvenileM",
           "Adult M = Juvenile F" = "centx_sexageadultM = centx_sexagejuvenileF",
           "Juvenile M = Juvenile F" = "centx_sexagejuvenileM = centx_sexagejuvenileF")
# Test hypotheses
h_bred <- hypothesis(d_yr_cent, v_bred)
# Store results
h_bred_tib <- h_bred$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)
```

```r
## Post-breeding
# Define hypotheses
v_post <- c("Adult F = Adult M" =
              "0 = centx_sexageadultM + centx_periodPost_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = centx_sexagejuvenileF + centx_periodPost_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = centx_sexagejuvenileM + centx_periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "centx_sexageadultM + centx_periodPost_breeding:sexageadultM = centx_sexagejuvenileM +
             centx_periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "centx_sexageadultM + centx_periodPost_breeding:sexageadultM = centx_sexagejuvenileF +
             centx_periodPost_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "centx_sexagejuvenileM + centx_periodPost_breeding:sexagejuvenileM = centx_sexagejuvenileF +
              centx_periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_post <- hypothesis(d_yr_cent, v_post)
# Store results
h_post_tib <- h_post$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_centx_onto <-
  full_join(h_sum_tib, h_pre_tib, by = "Hypothesis") %>%
  full_join(., h_bred_tib, by = "Hypothesis") %>%
  full_join(., h_post_tib, by = "Hypothesis")

### Latitude --------------------------------------------------------------------------------------------------------
## Summer
# Define hypotheses
v_sum <- c("Adult F = Adult M" =
             "0 = centy_sexageadultM + centy_periodSummer:sexageadultM",
          "Adult F = Juvenile F" =
            "0 = centy_sexagejuvenileF + centy_periodSummer:sexagejuvenileF",
          "Adult F = Juvenile M" =
            "0 = centy_sexagejuvenileM + centy_periodSummer:sexagejuvenileM",
          "Adult M = Juvenile M" =
            "centy_sexageadultM + centy_periodSummer:sexageadultM = centy_sexagejuvenileM +
             centy_periodSummer:sexagejuvenileM",
          "Adult M = Juvenile F" =
            "centy_sexageadultM + centy_periodSummer:sexageadultM = centy_sexagejuvenileF +
             centy_periodSummer:sexagejuvenileF",
          "Juvenile M = Juvenile F" =
            "centy_sexagejuvenileM + centy_periodSummer:sexagejuvenileM = centy_sexagejuvenileF +
             centy_periodSummer:sexagejuvenileF")
# Test hypotheses
h_sum <- hypothesis(d_yr_cent, v_sum)
# Store results
h_sum_tib <- h_sum$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Pre-breeding
# Define hypotheses
v_pre <- c("Adult F = Adult M" =
             "0 = centy_sexageadultM + centy_periodPre_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = centy_sexagejuvenileF + centy_periodPre_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = centy_sexagejuvenileM + centy_periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "centy_sexageadultM + centy_periodPre_breeding:sexageadultM = centy_sexagejuvenileM +
             centy_periodPre_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "centy_sexageadultM + centy_periodPre_breeding:sexageadultM = centy_sexagejuvenileF +
             centy_periodPre_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "centy_sexagejuvenileM + centy_periodPre_breeding:sexagejuvenileM = centy_sexagejuvenileF +
              centy_periodPre_breeding:sexagejuvenileF")
# Test hypotheses
h_pre <- hypothesis(d_yr_cent, v_pre)
# Store results
h_pre_tib <- h_pre$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Breeding
# Define hypotheses
v_bred <- c("Adult F = Adult M" = "0 = centy_sexageadultM ",
           "Adult F = Juvenile F" = "0 = centy_sexagejuvenileF",
           "Adult F = Juvenile M" = "0 = centy_sexagejuvenileM",
           "Adult M = Juvenile M" = "centy_sexageadultM = centy_sexagejuvenileM",
           "Adult M = Juvenile F" = "centy_sexageadultM = centy_sexagejuvenileF",
           "Juvenile M = Juvenile F" = "centy_sexagejuvenileM = centy_sexagejuvenileF")
# Test hypotheses
h_bred <- hypothesis(d_yr_cent, v_bred)
# Store results
h_bred_tib <- h_bred$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

## Post-breeding
# Define hypotheses
v_post <- c("Adult F = Adult M" =
              "0 = centy_sexageadultM + centy_periodPost_breeding:sexageadultM",
           "Adult F = Juvenile F" =
             "0 = centy_sexagejuvenileF + centy_periodPost_breeding:sexagejuvenileF",
           "Adult F = Juvenile M" =
             "0 = centy_sexagejuvenileM + centy_periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile M" =
             "centy_sexageadultM + centy_periodPost_breeding:sexageadultM = centy_sexagejuvenileM +
             centy_periodPost_breeding:sexagejuvenileM",
           "Adult M = Juvenile F" =
             "centy_sexageadultM + centy_periodPost_breeding:sexageadultM = centy_sexagejuvenileF +
             centy_periodPost_breeding:sexagejuvenileF",
           "Juvenile M = Juvenile F" =
             "centy_sexagejuvenileM + centy_periodPost_breeding:sexagejuvenileM = centy_sexagejuvenileF +
              centy_periodPost_breeding:sexagejuvenileF")
# Test hypotheses
h_post <- hypothesis(d_yr_cent, v_post)
# Store results
h_post_tib <- h_post$hypothesis %>%
  mutate(CI = paste("(",format(round(`CI.Lower`,2),nsmall = 2), " - ",
                    format(round(`CI.Upper`,2),2), ")", sep = "")) %>%
  dplyr::select(Hypothesis, Estimate, CI, `Evidence Ratio` = Evid.Ratio,
                `Posterior Probability` = Post.Prob, Star)

# Combine results from all tested hypotheses
h_all_centy_onto <-
  full_join(h_sum_tib, h_pre_tib, by = "Hypothesis") %>%
  full_join(., h_bred_tib, by = "Hypothesis") %>%
  full_join(., h_post_tib, by = "Hypothesis")
```

Individual differences

```
###########################################################################################################
#### Extract parameters from Mean and Dispersion models ###################################################
###########################################################################################################
## Extract the population coefficients for mean model
# Adult females
Baf_data <- tibble(location = "Center GSL",
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "adult F",
                   deploy.yr = mean(hr.data$deploy.yr),
                   id = "null")
# Extract the posterior draws of the linear predictor
Baf_all <- posterior_linpred(d_yr_cent, newdata = Baf_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Baf_centx <- Baf_all[,1,1]*(4/11) + Baf_all[,2,1]*(3/11) + Baf_all[,3,1]*(1/11) +
  Baf_all[,4,1]*(3/11)
Baf_centy <- Baf_all[,1,2]*(4/11) + Baf_all[,2,2]*(3/11) + Baf_all[,3,2]*(1/11) +
  Baf_all[,4,2]*(3/11)

# Adult males
Bam_data <- tibble(location = "Center GSL",
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "adult M",
                   deploy.yr = mean(hr.data$deploy.yr),
                   id = "null")
# Extract the posterior draws of the linear predictor
Bam_all <- posterior_linpred(d_yr_cent, newdata = Bam_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bam_centx <- Bam_all[,1,1]*(4/11) + Bam_all[,2,1]*(3/11) + Bam_all[,3,1]*(1/11) +
  Bam_all[,4,1]*(3/11)
Bam_centy <- Bam_all[,1,2]*(4/11) + Bam_all[,2,2]*(3/11) + Bam_all[,3,2]*(1/11) +
  Bam_all[,4,2]*(3/11)


# Juvenile females
Bjf_data <- tibble(location = "Center GSL",
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "juvenile F",
                   deploy.yr = mean(hr.data$deploy.yr),
                   id = "null")
# Extract the posterior draws of the linear predictor
Bjf_all <- posterior_linpred(d_yr_cent, newdata = Bjf_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bjf_centx <- Bjf_all[,1,1]*(4/11) + Bjf_all[,2,1]*(3/11) + Bjf_all[,3,1]*(1/11) +
  Bjf_all[,4,1]*(3/11)
Bjf_centy <- Bjf_all[,1,2]*(4/11) + Bjf_all[,2,2]*(3/11) + Bjf_all[,3,2]*(1/11) +
  Bjf_all[,4,2]*(3/11)

# Juvenile males
Bjm_data <- tibble(location = "Center GSL",
                   period = c("Summer","Pre_breeding", "Breeding", "Post_breeding"),
                   sexage = "juvenile M",
                   deploy.yr = mean(hr.data$deploy.yr),
                   id = "null")
# Extract the posterior draws of the linear predictor
Bjm_all <- posterior_linpred(d_yr_cent, newdata = Bjm_data, re.form = NA)
# Adjusting mean for the number of months each period last in a year
Bjm_centx <- Bjm_all[,1,'centx']*(4/11) + Bjm_all[,2,'centx']*(3/11) +
  Bjm_all[,3,'centx']*(1/11) + Bjm_all[,4,'centx']*(3/11)
Bjm_centy <- Bjm_all[,1,'centy']*(4/11) + Bjm_all[,2,'centy']*(3/11) +
  Bjm_all[,3,'centy']*(1/11) + Bjm_all[,4,'centy']*(3/11)

## Calculate the population intercept (average individual phenotype)
## for the mean model (B_pm) and the dispersion model (B_pv)
# Mean model (assuming equal representation of each sexage class)
B_pm_centx <- Baf_centx*(1/4) + Bam_centx*(1/4) + Bjf_centx*(1/4) + Bjm_centx*(1/4)
B_pm_centy <- Baf_centy*(1/4) + Bam_centy*(1/4) + Bjf_centy*(1/4) + Bjm_centy*(1/4)

# Dispersion model
## Extract the population coefficients for dispersion model
Bint_sd_centx <- as_draws_matrix(d_yr_cent, variable = "b_sigma_centx_Intercept")
Bint_sd_centy <- as_draws_matrix(d_yr_cent, variable = "b_sigma_centy_Intercept")
# Convert to variance scale scale
# brms uses residual standard deviation as the response variable for the dispersion model (ln(sd)).
# We need to convert to variance scale by multiplying by 2 for the dispersion model intercepts.
B_pv_centx <- 2 * Bint_sd_centx
B_pv_centy <- 2 * Bint_sd_centy

## Extract the random ID intercept variance components.
# Mean model
sigma_ID_m0_centx <- as_draws_matrix(d_yr_cent, variable = "sd_id__centx_Intercept")
sigma_ID_m0_centy <- as_draws_matrix(d_yr_cent, variable = "sd_id__centy_Intercept")

# Dispersion model
sigma_ID_sd0_centx <- as_draws_matrix(d_yr_cent, variable = "sd_id__sigma_centx_Intercept")
sigma_ID_sd0_centy <- as_draws_matrix(d_yr_cent, variable = "sd_id__sigma_centy_Intercept")

# Convert to variance scale
sigma_ID_v0_centx <- 2 * sigma_ID_sd0_centx
sigma_ID_v0_centy <- 2 * sigma_ID_sd0_centy

## Repeat process for the random deploy year intercept variance components.
# Mean model
sigma_deploy.yr_m0_centx <- as_draws_matrix(d_yr_cent, variable = "sd_deploy.yr__centx_Intercept")
sigma_deploy.yr_m0_centy <- as_draws_matrix(d_yr_cent, variable = "sd_deploy.yr__centy_Intercept")

# Dispersion model
sigma_deploy.yr_sd0_centx <- as_draws_matrix(d_yr_cent, variable = "sd_deploy.yr__sigma_centx_Intercept")
sigma_deploy.yr_sd0_centy <- as_draws_matrix(d_yr_cent, variable = "sd_deploy.yr__sigma_centy_Intercept")

# Convert to variance scale
sigma_deploy.yr_v0_centx <- 2 * sigma_deploy.yr_sd0_centx
sigma_deploy.yr_v0_centy <- 2 * sigma_deploy.yr_sd0_centy


## Calculate the fixed effect variance for the mean model (var_fixed_m) and dispersion model (var_fixed_v).
# Mean model
fixed_pred_m <- posterior_linpred(d_yr_cent,
                                  newdata = hr.data,
                                  re.form = NA)

var_fixed_m_centx <- sapply(1:length(fixed_pred_m[,1,'centx']), function(x) var(fixed_pred_m[x,,'centx']))
var_fixed_m_centy <- sapply(1:length(fixed_pred_m[,1,'centy']), function(x) var(fixed_pred_m[x,,'centy']))

# Dispersion model
# Intercept only for sigma so no variance for fixed effects
var_fixed_sd_centx <- 0
var_fixed_sd_centy <- 0

# Convert to variance scale
# Again, brms uses residual standard deviation as the response variable for the dispersion model (ln(sd)).
# Variance on the ln(sd) scale can be converted to variance on the ln(var) scale by multipling by 4.
var_fixed_v_centx <- 4 * var_fixed_sd_centx
var_fixed_v_centy <- 4 * var_fixed_sd_centy

####################################################################################################################
```

```
### Mean Model - Repeatability and coefficient of variation ########################################################
####################################################################################################################
## Calculate average residual variance
# summed variance components
var_within_exp_centx <- func_sum_var(vars = data.frame(sigma_ID_v0_centx^2, sigma_deploy.yr_v0_centx^2))
var_within_exp_centy <- func_sum_var(vars = data.frame(sigma_ID_v0_centy^2, sigma_deploy.yr_v0_centy^2))
# conversion back from log scale
sigma_w_centx <- exp(B_pv_centx + var_within_exp_centx/2)
sigma_w_centy <- exp(B_pv_centy + var_within_exp_centy/2)

## Calculate total phenotypic variance
# Total phenotypic variance (with fixed effects)
var_p_centx <- func_sum_var(vars = data.frame(sigma_ID_m0_centx^2,
                                              sigma_deploy.yr_m0_centx^2,
                                              var_fixed_m_centx,
                                              sigma_w_centx^2))

var_p_centy <- func_sum_var(vars = data.frame(sigma_ID_m0_centy^2,
                                              sigma_deploy.yr_m0_centy^2,
                                              var_fixed_m_centy,
                                              sigma_w_centy^2))

# 'Adjusted' phenotypic variance (without fixed effects)
var_p_nofixef_centx <- func_sum_var(vars = data.frame(sigma_ID_m0_centx^2,
                                                      sigma_deploy.yr_m0_centx^2,
                                                      sigma_w_centx^2))

var_p_nofixef_centy <- func_sum_var(vars = data.frame(sigma_ID_m0_centy^2,
                                                      sigma_deploy.yr_m0_centy^2,
                                                      sigma_w_centy^2))

## Calculate Repeatability
# Agreement repeatability (fixed effects var in denominator)
Rp_mu_centx <- sigma_ID_m0_centx^2/var_p_centx
Rp_mu_centy <- sigma_ID_m0_centy^2/var_p_centy

# Adjusted repeatability (fixed effects var excluded from denominator)
Rp_mu_adj_centx <- sigma_ID_m0_centx^2/var_p_nofixef_centx
Rp_mu_adj_centy <- sigma_ID_m0_centy^2/var_p_nofixef_centy

## Calculate Coefficient of variation
CV_mu_centx <- sigma_ID_m0_centx/B_pm_centx
CV_mu_centy <- sigma_ID_m0_centy/B_pm_centy

####################################################################################################################
### Dispersion Model - Repeatability and coefficient of variation ##################################################
####################################################################################################################
## Total variance in residual variance
# On log-normal scale:
var_residvar_exp_centx <- func_sum_var(vars = data.frame(sigma_ID_v0_centx^2,
                                                   sigma_deploy.yr_v0_centx^2))

var_residvar_exp_centy <- func_sum_var(vars = data.frame(sigma_ID_v0_centy^2,
                                                   sigma_deploy.yr_v0_centy^2))

# Converting back to same scale as mean model:
var_residvar_centx <- (exp(var_residvar_exp_centx) - 1) * exp(2 * B_pv_centx + var_residvar_exp_centx)
var_residvar_centy <- (exp(var_residvar_exp_centy) - 1) * exp(2 * B_pv_centy + var_residvar_exp_centy)

## Total variance in phenotypic variance (repeatability denominator)
total_var_var_centx <- 2 * var_p_centx^2 + 3 * var_residvar_centx
total_var_var_centy <- 2 * var_p_centy^2 + 3 * var_residvar_centy

## Getting variance in individual component (repeatability numerator) through the
## preservation of the proportionality (i.e. ratio method; O'dea et al. 2022)
var_ID_centx <- var_residvar_centx * (sigma_ID_v0_centx^2/var_residvar_exp_centx)
```

```r
var_ID_centy <- var_residvar_centy * (sigma_ID_v0_centy^2/var_residvar_exp_centy)

## Calculate repeatability for the dispersion model
Rp_var_centx <- var_ID_centx/total_var_var_centx
Rp_var_centy <- var_ID_centy/total_var_var_centy

## Calculate coefficient of variation for the dispersion model
CV_var_centx <- sqrt(var_ID_centx)/sigma_w_centx^2
CV_var_centy <- sqrt(var_ID_centy)/sigma_w_centy^2
```

Figures

Figure 1 - Study system

```r
# A) Deployment sites map -----------------------------------------------------------------------------

# Generate deployment sites tibble
#world <- ne_countries(scale = "medium", returnclass = "sf")
world <- ne_countries(scale = "large", returnclass = "sf")

sites <- tibble(
  sites = c("St George's Bay", "Anticosti Island", "Métis-sur-Mer",
            "Amet Island", "Kouchibouguac", "Brion Island"),
  longitude = c(-61.690711993886, -62.99424426648467, -68.0330995267766,
                -63.17714575179163, -64.86289734875632, -61.49320012194406),
  latitude = c(45.78560555624806, 49.52212432174339, 48.68187699980851,
               45.835499821072546, 46.76743173640179, 47.78602556317871),
  # Adjust the position of the labels on the x axis of the graph
  nudgex = c(0.3,2.3,1.8,
             -1.8,-2.0,0.0),
  # Adjust the position of the labels on the y axis of the graph
  nudgey = c(-0.8,0.2,-0.1,
             -0.2,0.0,0.6),
  # Categorize deployment sites by region
  region = case_when(sites == "Brion Island" ~ "Center GSL",
                     sites == "Anticosti Island" ~ "North GSL",
                     sites == "Métis-sur-Mer" ~ "North GSL",
                     sites == "St George's Bay" ~ "South GSL",
                     sites == "Amet Island" ~ "South GSL",
                     sites == "Kouchibouguac" ~ "South GSL",
                     TRUE ~ "Scotian Shelf")) %>%
  # Transform into factor and reorder the 'region' column
  mutate(region = factor(region,
                         levels = c("North GSL", "Center GSL", "South GSL"))) %>%
  arrange(region)

sites_sf <- st_as_sf(sites, coords = c("longitude", "latitude"),
                     crs = 4326, agr = "constant")

bbox <- st_bbox(sites_sf)

# Deployment sites
deployment_sites <- ggplot(data = world) +
  geom_sf(fill= "darkgrey") +
  geom_sf(data = sites_sf, aes(color=region, fill = region), size = 1.5) +
  geom_sf_label_repel(data = sites_sf, aes(label = sites, color=region),
                      force = 0, min.segment.length = 0,
                      size = 2,
                      nudge_x = sites$nudgex,
                      nudge_y = sites$nudgey) +
  coord_sf(crs = st_crs(4326),
           xlim = c(bbox['xmin']-0.3, bbox['xmax']+3.5),
           ylim = c(bbox['ymin']-2, bbox['ymax']+2)) +
  annotation_north_arrow(location = "br", which_north = "true",
                         height = unit(1, "cm"),
                         width = unit(1, "cm"),
                         pad_x = unit(0.05, "cm"),
                         pad_y = unit(0.15, "cm"),
                         style = north_arrow_nautical()) +
  labs(x = "Longitude", y = "Latitude",
       color = "Deployment location", fill = "Deployment location") +
  scale_color_viridis_d(direction = 1, option = "magma", begin = 0.1, end = 0.85) +
  scale_fill_viridis_d(direction = 1, option = "magma", begin = 0.1, end = 0.85,) +
  guides(color = guide_legend(override.aes = list(linetype = c(0, 0, 0)),
                              title.position = "top",
                              direction = "vertical"),
         fill = guide_legend(override.aes = list(fill = viridis_pal(
           direction = 1, option = "magma", begin = 0.1, end = 0.85)(3)),
           title.position = "top",
           direction = "vertical")) +
  theme_minimal() +
  theme(legend.position = "top",
        legend.box.margin = margin(t=-8, r=0, b=-15, l=-70),
        plot.margin = margin(t=0, r=0, b=0, l=-25),
        panel.border = element_rect(colour = "black", fill=NA, size=1))

# Inset map of canada
canada_map <- ggplot(data = filter(world, sovereignt == "Canada")) +
  geom_sf(fill= "darkgrey") +
  geom_rect(aes(xmin = bbox['xmin']-2,
                xmax = bbox['xmax']+5,
                ymin = bbox['ymin']-3,
                ymax = bbox['ymax']+3),
            color='red', alpha=0, linewidth = 0.5) +
  cowplot::theme_nothing() +
  theme(panel.background=element_rect(fill="white"),
        panel.border = element_rect(colour = "black", fill=NA, size=1))

deployment_map <- deployment_sites +
  annotation_custom(ggplotGrob(canada_map),
                    xmin = -68.7, xmax = -66,
                    ymin = 48, ymax = 53.8)

# B) Tag coverage -------------------------------------------------------------------------------------

# Load and tidy metadata
cov_GSL <-
  # Import metadata file
  Metadata %>%
  dplyr::select(id, year, start, end) %>%
  # Keep only the individuals that are in the data
  filter(id %in% unique(hr.data$id)) %>%
  # Change Feb 29 to Feb 28, because Feb 29 cause a bug.
  mutate(end = if_else(end == as_date("2004-02-29"), as_date("2004-02-28"), end)) %>%
  # Separate start column into year, month, date columns
  separate(start, c("start.year","start.month","start.day", sep = "-")) %>%
  # Same as above for end column
  separate(end, c("end.year","end.month","end.day", sep = "-")) %>%
  mutate(
    # Calculate difference between end and deployment year
    year.diff.start = as.numeric(start.year) - as.numeric(year),
    year.diff.end = as.numeric(end.year) - as.numeric(year),
    # Set start year as 2020 so all the tags are aligned in the graph
    start.year = 2020 + year.diff.start,
    # Set end year as 2020 + year.diff so tags that ended in a different year
    # also end in a different year on the graph
    end.year = 2020 + year.diff.end) %>%
  # Reunite start year, month and day columns
  unite("start.date", start.year:start.day, sep = "-", remove = FALSE) %>%
  # Reunite end year, month and day columns
  unite("end.date", end.year:end.day, sep = "-", remove = FALSE) %>%
  mutate(
    # Create start column in date format
    start = as.POSIXct(start.date, format = "%Y-%m-%d"),
    # Create start column in date format
    end = as.POSIXct(end.date, format = "%Y-%m-%d"),
    # Limit end date to April 30th as home range were not calculate in May
    end = if_else(end > as.POSIXct("2021-04-30", format = "%Y-%m-%d"),
                  as.POSIXct("2021-04-30", format = "%Y-%m-%d"),
                  end)
  )

# Create plot
coverage <- ggplot(cov_GSL,
                   aes(x=factor(id, levels = id[order(year)]),
                       colour=as.factor(year))) +
  labs(x = "Id", y = "Date") +
  scale_y_datetime(date_breaks = "1 month",
                   minor_breaks = "1 year",
                   labels = date_format("%b"),
                   limits = as.POSIXct(strptime(c("2020-06-01","2021-04-30"), format = "%Y-%m-%d")) ) +
  geom_linerange(aes(ymin = start, ymax = end),
                 linewidth = 0.75,
                 key_glyph='rect') +
  geom_hline(yintercept = as.POSIXct(
    strptime(
      seq(ymd("2020-01-01"), ymd("2021-06-01"), by = "1 month"),
      format = "%Y-%m-%d")),
    size = 0.1) +
  guides(colour = guide_legend(reverse = TRUE,
                               title.position = "top")) +
  labs(colour = "Deployment year") +
  scale_color_viridis_d(direction = 1, option = "mako") +
  coord_flip() +
  theme_minimal() +
  theme(axis.text.y=element_blank(),
        panel.grid.major.y=element_blank(),
        text = element_text(size = 20),
        legend.position = "top",
        legend.key.height = unit(0.2, 'cm'),
        legend.key.width = unit(0.6, 'cm'),
        legend.box.margin = margin(t=-8, r=0, b=-10, l=0))

# Combine ---------------------------------------------------------------------------------------------

Fig_1 <- plot_grid(
  deployment_map +
    theme(
      text = element_text(size=10),
      legend.text = element_text(size=8)),

  coverage +
    theme(text = element_text(size = 10),
          legend.text = element_text(size=8)),

  labels = "AUTO", align = "h",
  hjust = c(-0.5, -0.5))

ggsave(filename = "Figures/Figure_1.tiff",
       plot = Fig_1,
       units = "cm",
       dpi = 300,
       height = 10,
       width = 16.5,
        bg = "white")
```

Figure 2 - Range and distribution of shape index

```
# Generate components  --------------------------------------------------------------------------------
# Determine Quantile
loc_min <- filter( fit.month.hr,
                  shapeindex == quantile( fit.month.hr$shapeindex)[1])
loc_q1 <- filter( fit.month.hr,
                 abs(shapeindex - quantile( fit.month.hr$shapeindex)[2]) == min(abs(shapeindex - quantile( fit.month.hr$shape
loc_q2 <- filter( fit.month.hr,
                 abs(shapeindex - quantile( fit.month.hr$shapeindex)[3]) == min(abs(shapeindex - quantile( fit.month.hr$shape
loc_q3 <- filter( fit.month.hr,
                 abs(shapeindex - quantile( fit.month.hr$shapeindex)[4]) == min(abs(shapeindex - quantile( fit.month.hr$shape
loc_max <- filter( fit.month.hr,
                  shapeindex == quantile( fit.month.hr$shapeindex)[5])

# Generate plots per quantile
loc_gg_min <- ggplot() +
  geom_sf(data = loc_min$poly95[[1]]$geometry,
          fill = "#278f8f", alpha = 0.5, color = "#278f8f") +
  theme_void()

loc_gg_q1 <- ggplot() +
  geom_sf(data = loc_q1$poly95[[1]]$geometry,
          fill = "#278f8f", alpha = 0.5, color = "#278f8f") +
  theme_void()

loc_gg_q2 <- ggplot() +
  geom_sf(data = loc_q2$poly95[[1]]$geometry,
          fill = "#278f8f", alpha = 0.5, color = "#278f8f") +
  theme_void()

loc_gg_q3 <- ggplot() +
  geom_sf(data = loc_q3$poly95[[1]]$geometry,
          fill = "#278f8f", alpha = 0.5, color = "#278f8f") +
  theme_void()

loc_gg_max <- ggplot() +
  geom_sf(data = loc_max$poly95[[1]]$geometry,
          fill = "#278f8f", alpha = 0.5, color = "#278f8f") +
  theme_void()

# Shape index per quantile
Min <- text_grob(format(round(quantile( fit.month.hr$shapeindex)[1], digits = 2), nsmall=2),
                 size = 10)

Q1 <- text_grob(format(round(loc_q1$shapeindex[1], digits = 2), nsmall=2),
                size = 10)

Median <- text_grob(format(round(loc_q2$shapeindex[1], digits = 2), nsmall=2),
                    size = 10)

Q3 <- text_grob(format(round(loc_q3$shapeindex[1], digits = 2), nsmall=2),
                size = 10)

Max <- text_grob(format(round(quantile( fit.month.hr$shapeindex)[5], digits = 2), nsmall =2),
                 size = 10)

# Generate figure --------------------------------------------------------------------------------
Fig_2 <- plot_grid(
  annotate_figure(loc_gg_min, top = "Min.", bottom = Min),
```

```r
  annotate_figure(loc_gg_q1, top = "1st Qu.", bottom = Q1),

  annotate_figure(loc_gg_q2, top = "Median", bottom = Median),

  annotate_figure(loc_gg_q3, top = "3rd Qu.", bottom = Q3),

  annotate_figure(loc_gg_max, top = "Max.", bottom = Max),

  ggplot( fit.month.hr, aes(x = shapeindex)) +
    geom_histogram(binwidth = 0.2) +
    scale_x_continuous(expand = c(0.01, 0.01)) +
    scale_y_continuous(expand = c(0.01, 0.01)) +
    labs(x="Shape index", y = "Count") +
    theme_classic() +
    theme(text = element_text(size = 8),
          plot.margin = unit(c(0,0.2,0.2,0), "lines")),

  ncol = 3, labels = "AUTO", label_y = 1.02,
  label_x = c(-0.02, -0.02, -0.03,
              -0.02, -0.02, -0.03),
  align = "h")

ggsave(filename = "Figures/Figure_2.tiff",
       plot = Fig_2,
       units = "cm",
       dpi = 300,
       height = 8,
       width = 10,
       bg = "white")
```

Figure 3 - Ontological and seasonal variation

```r
# Generate data ----------------------------------------------------------------

## Area

# Store model predictions
sexage_area <- conditional_effects(d_yr_area, effects = "sexage:period")
# Data extraction and formatting
sexage_area_dat <- sexage_area$`sexage:period` %>% convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period))

# Generate annotation tibble based on results from post-hoc comparisons
# Between periods of the year, within sex-age class (Table S1)
annotation_area_seas <- data.frame(
  sexage = c(rep(c("adult F"),4),rep(c("adult M"),4),
             rep(c("juvenile F"),4),rep(c("juvenile M"),4)),
  period = rep(c("Summer","Pre-breeding","Breeding","Post-breeding"), 4),
  label_seas = c("a", "b","b","b",
                 "a", "ab","b","c",
                 "a", "b","b","b",
                 "a", "a","a","a"))
# Between sex-age class, within periods of the year (Table S2)
annotation_area_onto <- data.frame(
  sexage = rep(c("adult F","adult M","juvenile F","juvenile M"), 4),
  period = c(rep(c("Summer"),4),rep(c("Pre-breeding"),4),
             rep(c("Breeding"),4),rep(c("Post-breeding"),4)),
  label_onto = c("A", "B","A","B",
                 "A", "A","A","A",
                 "A", "B","B","AB",
                 "A", "B","C","C"))
```

```r
# Combine data and annotations
sexage_area_dat <- sexage_area_dat %>%
  left_join(., annotation_area_seas, by = c('sexage', 'period')) %>%
  left_join(., annotation_area_onto, by = c('sexage', 'period')) %>%
  mutate(period = factor(period, levels = c("Summer", "Pre-breeding",
                                            "Breeding","Post-breeding")))

## Shape

# Store model predictions
sexage_shape <- conditional_effects(d_yr_shape, effects = "sexage:period")
# Data extraction and formatting
sexage_shape_dat <- sexage_shape$`sexage:period` %>% convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period))

# Generate annotation tibble based on results from post-hoc comparisons
# Between periods of the year, within sex-age class (Table S1)
annotation_shape_seas <- data.frame(
  sexage = c(rep(c("adult F"),4),rep(c("adult M"),4),
             rep(c("juvenile F"),4),rep(c("juvenile M"),4)),
  period = rep(c("Summer","Pre-breeding","Breeding","Post-breeding"), 4),
  label_seas = c("a", "b","c","c",
                 "a", "a","b","c",
                 "a", "b","c","c",
                 "a", "a","a","b"))
# Between sex-age class, within periods of the year (Table S2)
annotation_shape_onto <- data.frame(
  sexage = rep(c("adult F","adult M","juvenile F","juvenile M"), 4),
  period = c(rep(c("Summer"),4),rep(c("Pre-breeding"),4),
             rep(c("Breeding"),4),rep(c("Post-breeding"),4)),
  label_onto = c("A", "B","A","B",
                 "A", "A","A","A",
                 "A", "A","A","A",
                 "A", "B","AB","AB"))

# Combine data and annotations
sexage_shape_dat <- sexage_shape_dat %>%
  left_join(., annotation_shape_seas, by = c('sexage', 'period')) %>%
  left_join(., annotation_shape_onto, by = c('sexage', 'period')) %>%
  mutate(period = factor(period, levels = c("Summer", "Pre-breeding",
                                            "Breeding","Post-breeding")))

## Centroid latitude (y)

# Store model predictions
sexage_centy <- conditional_effects(d_yr_cent, effects = "sexage:period",resp = "centy")
# Data extraction and formatting
sexage_centy_dat <- sexage_centy$`centy.centy_sexage:period` %>%
  convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period))

# Generate annotation tibble based on results from post-hoc comparisons
# Between periods of the year, within sex-age class (Table S1)
annotation_centy_seas <- data.frame(
  sexage = c(rep(c("adult F"),4),rep(c("adult M"),4),
             rep(c("juvenile F"),4),rep(c("juvenile M"),4)),
  period = rep(c("Summer","Pre-breeding","Breeding","Post-breeding"), 4),
  label_seas = c("a", "b","c","c",
                 "a", "a","b","b",
                 "a", "a","a","a",
                 "ab", "a","ab","c"))
# Between sex-age class, within periods of the year (Table S2)
```

```
annotation_centy_onto <- data.frame(
  sexage = rep(c("adult F","adult M","juvenile F","juvenile M"), 4),
  period = c(rep(c("Summer"),4),rep(c("Pre-breeding"),4),
             rep(c("Breeding"),4),rep(c("Post-breeding"),4)),
  label_onto = c("A", "A","A","A",
                 "A", "A","A","A",
                 "A", "A","B","B",
                 "A", "A","B","A"))

# Combine data and annotations
sexage_centy_dat <- sexage_centy_dat %>%
  left_join(., annotation_centy_seas, by = c('sexage', 'period')) %>%
  left_join(., annotation_centy_onto, by = c('sexage', 'period')) %>%
  mutate(period = factor(period, levels = c("Summer", "Pre-breeding",
                                            "Breeding","Post-breeding")))

## Centroid longitude (x)

# Store model predictions
sexage_centx <- conditional_effects(d_yr_cent, effects = "sexage:period",resp = "centx")
# Data extraction and formatting
sexage_centx_dat <- sexage_centx$`centx.centx_sexage:period` %>%
  convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period))

# Generate annotation tibble based on results from post-hoc comparisons
# Between periods of the year, within sex-age class (Table S1)
annotation_centx_seas <- data.frame(
  sexage = c(rep(c("adult F"),4),rep(c("adult M"),4),
             rep(c("juvenile F"),4),rep(c("juvenile M"),4)),
  period = rep(c("Summer","Pre-breeding","Breeding","Post-breeding"), 4),
  label_seas = c("a", "a","b","ab",
                 "a", "b","c","d",
                 "a", "a","a","b",
                 "a", "a","b","b"))
# Between sex-age class, within periods of the year (Table S2)
annotation_centx_onto <- data.frame(
  sexage = rep(c("adult F","adult M","juvenile F","juvenile M"), 4),
  period = c(rep(c("Summer"),4),rep(c("Pre-breeding"),4),
             rep(c("Breeding"),4),rep(c("Post-breeding"),4)),
  label_onto = c("A", "A","A","A",
                 "A", "A","A","A",
                 "AB", "A","B","AB",
                 "A", "B","A","A"))

# Combine data and annotations
sexage_centx_dat <- sexage_centx_dat %>%
  left_join(., annotation_centx_seas, by = c('sexage', 'period')) %>%
  left_join(., annotation_centx_onto, by = c('sexage', 'period')) %>%
  mutate(period = factor(period, levels = c("Summer", "Pre-breeding",
                                            "Breeding","Post-breeding")))

# Generate figure --------------------------------------------------------------

# Area
sexage_area_gg <-
  ggplot(data=sexage_area_dat, aes(x=sexage,y=estimate__)) +
  geom_point(aes(color = period), size = 2, position = position_dodge(width = 0.8)) +
  geom_errorbar(aes(y=estimate__, ymin= lower__, ymax= upper__, color = period),
                position = position_dodge(width = 0.8)) +
  geom_text(aes(x = sexage, y = upper__, label = label_seas, group = period),
            position = position_dodge(width = 0.8), vjust = -0.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.5) +
  geom_text(aes(x = sexage, y = lower__, label = label_onto, color = period),
            position = position_dodge(width = 0.8), vjust = 1.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.6) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Sex-age class", y = bquote("Area "~(km^2)),
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr()+
  theme(text = element_text(size=10))

#sexage_area_gg

# Shape
sexage_shape_gg <- ggplot(data=sexage_shape_dat, aes(x=sexage,y=estimate__)) +
  geom_point(aes(color = period), size = 2, position = position_dodge(width = 0.8)) +
  geom_errorbar(aes(y=estimate__, ymin= lower__, ymax= upper__, color = period),
                position = position_dodge(width = 0.8)) +
  geom_text(aes(x = sexage, y = upper__, label = label_seas, group = period),
            position = position_dodge(width = 0.8), vjust = -0.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.5) +
  geom_text(aes(x = sexage, y = lower__, label = label_onto, color = period),
            position = position_dodge(width = 0.8), vjust = 1.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.6) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Sex-age class", y = "Shape index",
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr()+
  theme(text = element_text(size=10))

#sexage_shape_gg

# Centroid latitude (centy)
sexage_centy_gg <- ggplot(data=sexage_centy_dat, aes(x=sexage,y=estimate__)) +
  geom_point(aes(color = period), size = 2, position = position_dodge(width = 0.8)) +
  geom_errorbar(aes(y=estimate__, ymin= lower__, ymax= upper__, color = period),
                position = position_dodge(width = 0.8)) +
  geom_text(aes(x = sexage, y = upper__, label = label_seas, group = period),
            position = position_dodge(width = 0.8), vjust = -0.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.5) +
  geom_text(aes(x = sexage, y = lower__, label = label_onto, color = period),
            position = position_dodge(width = 0.8), vjust = 1.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.6) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Sex-age class", y = "Centroid latitude",
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr()+
  theme(text = element_text(size=10))

#sexage_centy_gg

# Centroid longitude (centx)
sexage_centx_gg <- ggplot(data=sexage_centx_dat, aes(x=sexage,y=estimate__)) +
  geom_point(aes(color = period), size = 2, position = position_dodge(width = 0.8)) +
  geom_errorbar(aes(y=estimate__, ymin= lower__, ymax= upper__, color = period),
                position = position_dodge(width = 0.8)) +
  geom_text(aes(x = sexage, y = upper__, label = label_seas, group = period),
            position = position_dodge(width = 0.8), vjust = -0.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.5) +
  geom_text(aes(x = sexage, y = lower__, label = label_onto, color = period),
            position = position_dodge(width = 0.8), vjust = 1.2, lineheight = .5,
            show.legend = F, size = 3, alpha = 0.6) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Sex-age class", y = "Centroid longitude",
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr()+
  theme(text = element_text(size=10))

#sexage_centx_gg

# Legend
legend_gg <- ggpubr::get_legend(sexage_area_gg +
                          theme(legend.title.align=0.5,
                                legend.box.margin = margin(t=-10, r=0, b=0, l=0)))
# Figure
Fig_3 <- plot_grid(
  plot_grid(

    sexage_area_gg +
      theme(legend.position="none",
            text = element_text(size=8)),

    sexage_shape_gg +
      theme(legend.position="none",
            text = element_text(size=8)),

    sexage_centy_gg +
      theme(legend.position="none",
            text = element_text(size=8)),

    sexage_centx_gg +
      theme(legend.position="none",
            text = element_text(size=8)),

    ncol = 2, align = "vh", labels = "AUTO"),

  legend_gg,

  nrow = 2, rel_heights = c(1, .1))

ggsave(filename = "Figures/Figure_3.tiff",
       plot = Fig_3,
       units = "cm",
       dpi = 300,
       height = 16.5,
       width = 16.5,
       bg = "white")
```

Figure 4 - Temporal change

```
# Generate data ---------------------------------------------------------------------------------------

# Area
yr_area <- conditional_effects(d_yr_area, effects = "deploy.yr:period")
yr_area_dat <- yr_area$`deploy.yr:period` %>%
  convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period)) %>%
  mutate(period = factor(period,
                         levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding")))

# Shape
yr_shape <- conditional_effects(d_yr_shape, effects = "deploy.yr:period")
yr_shape_dat <- yr_shape$`deploy.yr:period` %>%
  convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period)) %>%
  mutate(period = factor(period,
                         levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding")))

# Centroid longitude (cent.x)
yr_centx <- conditional_effects(d_yr_cent, effects = "deploy.yr:period",
                                resp = "centx")
yr_centx_dat <- yr_centx$`centx.centx_deploy.yr:period` %>%
  convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period)) %>%
  mutate(period = factor(period,
                         levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding")))

# Centroid latitude (cent.y)
yr_centy <- conditional_effects(d_yr_cent, effects = "deploy.yr:period",
                                resp = "centy")
yr_centy_dat <- yr_centy$`centy.centy_deploy.yr:period` %>%
  convert(chr(period)) %>%
  mutate(period = case_when(period == "Pre_breeding" ~ "Pre-breeding",
                            period == "Post_breeding" ~ "Post-breeding",
                            TRUE ~ period)) %>%
  mutate(period = factor(period,
                         levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding")))

# Generate figure -------------------------------------------------------------------------------------

yr_area_gg <- ggplot(data=yr_area_dat, aes(x=deploy.yr,y=estimate__)) +
  geom_line(aes(colour = period), linewidth = 1) +
  geom_ribbon(aes(y=estimate__, ymin= lower__, ymax= upper__, fill = period),
              alpha = 0.2) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Year", y = bquote("Area "~(km^2)),
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr() +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        text = element_text(size=8)) +
  facet_wrap(~factor(period,
                     levels = c("Summer", "Pre-breeding",
                                "Breeding", "Post-breeding")),
             drop = F)

yr_shape_gg <- ggplot(data=yr_shape_dat, aes(x=deploy.yr,y=estimate__)) +
  geom_line(aes(colour = period), linewidth = 1) +
  geom_ribbon(aes(y=estimate__, ymin= lower__, ymax= upper__, fill = period),
              alpha = 0.2)+
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Year", y = "Shape index",
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr() +
```

```r
  theme(legend.position="none",
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        text = element_text(size=8)) +
  facet_wrap(~factor(period,
                     levels = c("Summer", "Pre-breeding",
                                "Breeding", "Post-breeding")),
             drop = F)

yr_centx_gg <- ggplot(data=yr_centx_dat, aes(x=deploy.yr,y=estimate__)) +
  geom_line(aes(colour = period), linewidth = 1)+
  geom_ribbon(aes(y=estimate__, ymin= lower__, ymax= upper__, fill = period),
              alpha = 0.2) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Year", y = "Centroid longitude",
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr()+
  theme(legend.position="none",
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        text = element_text(size=8)) +
  facet_wrap(~factor(period,
                     levels = c("Summer", "Pre-breeding",
                                "Breeding", "Post-breeding")),
             drop = F)

yr_centy_gg <- ggplot(data=yr_centy_dat, aes(x=deploy.yr,y=estimate__)) +
  geom_line(aes(colour = period), linewidth = 1)+
  geom_ribbon(aes(y=estimate__, ymin= lower__, ymax= upper__, fill = period),
              alpha = 0.2) +
  scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
  guides(color = guide_legend(title.position = "top"),
         fill = guide_legend(title.position = "top")) +
  labs(x = "Year", y = "Centroid latitude",
       fill = "Period of the year", color = "Period of the year") +
  theme_pubr() +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        text = element_text(size=8)) +
  facet_wrap(~factor(period,
                     levels = c("Summer", "Pre-breeding",
                                "Breeding", "Post-breeding")),
             drop = F)

# Legend
legend_gg <- ggpubr::get_legend(ggplot(data=yr_area_dat, aes(x=deploy.yr,y=estimate__)) +
                          geom_line(aes(colour = period), linewidth = 1) +
                          geom_ribbon(aes(y=estimate__, ymin= lower__, ymax= upper__, fill = period),
                                      alpha = 0.2) +
                          scale_color_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
                          scale_fill_viridis_d(direction = 1, begin = 0.1, end = 0.9, option = "mako") +
                          guides(color = guide_legend(title.position = "top"),
                                 fill = guide_legend(title.position = "top")) +
                          labs(x = "Year", y = bquote("Area "~(km^2)),
                               fill = "Period of the year", color = "Period of the year") +
                          theme_pubr()+
                          theme(text = element_text(size=10)) +
                          theme(legend.title.align=0.5,
                                legend.box.margin = margin(t=-15, r=0, b=0, l=0)))

# Figure
Fig_4 <- plot_grid(
  plot_grid(
    yr_area_gg,

    yr_shape_gg,

    yr_centy_gg,

    yr_centx_gg,

    ncol = 2, align = "vh", labels = "AUTO"),

  legend_gg,

  nrow = 2, rel_heights = c(1, .1))

ggsave(filename = "Figures/Figure_4.tiff",
       plot = Fig_4,
       units = "cm",
       dpi = 300,
       height = 16.5,
       width = 16.5,
       bg = "white")
```

Figure 5 - Home ranges maps

```
## Import and prepare data ---------------------------------------------------------------------------
# Load world map data
world <- ne_countries(scale = "medium", returnclass = "sf")

# Prepare home range data
hr.poly95 <- fit.month.hr %>%
  dplyr::select(id, deploy.yr, month, year, poly95)

# Simple features data formatting
sf_hr <- st_transform(as(hr.poly95[1,]$poly95[[1]], "sf") %>%
                        mutate(id = hr.poly95[1,]$id,
                               month = hr.poly95[1,]$month,
                               deploy.yr = hr.poly95[1,]$deploy.yr,
                               period = case_when(month %in% c(1) ~ "Breeding",
                                                  month %in% c(2,3,4) ~ "Post-breeding",
                                                  month %in% c(6,7,8,9) ~ "Summer",
                                                  month %in% c(10,11,12) ~ "Pre-breeding")),
                      crs = st_crs(4326))

for (i in seq(2,length(hr.poly95$id))) {

  sf_hr_idmonth <- st_transform(as(hr.poly95[i,]$poly95[[1]], "sf") %>%
                                  mutate(id = hr.poly95[i,]$id,
                                         month = hr.poly95[i,]$month,
                                         deploy.yr = hr.poly95[i,]$deploy.yr,
                                         period = case_when(month %in% c(1) ~ "Breeding",
                                                            month %in% c(2,3,4) ~ "Post-breeding",
                                                            month %in% c(6,7,8,9) ~ "Summer",
                                                            month %in% c(10,11,12) ~ "Pre-breeding")),
                                crs = st_crs(4326))

  sf_hr <- rbind(sf_hr, sf_hr_idmonth)

}

## Setting map theme ---------------------------------------------------------------------------------
# Set bounding box for maps
bbox <- st_bbox(sf_hr)
```

```r
# Set color theme for pre and post reproduction periods
period_fill <- setNames(object = c(scales::viridis_pal(direction = 1, begin = 0.1,end = 0.9, option = "mako")(5)),
                        nm = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))

# ggplot theme for graph
map_theme <- theme_minimal() +
  theme(panel.background = element_rect(fill="white", color = NA),
        plot.background = element_rect(fill="white", color = NA),
        plot.margin = unit(c(3,3,3,3), 'pt'),
        plot.title = element_text(hjust = 0.5),
        text = element_text(size=10))

## Deployment sites ----------------------------------------------------------------------------------
sites <- tibble(
  sites = c("St George's Bay", "Anticosti Island", "Métis-sur-Mer", "Bic Island",
            "Amet Island", "Kouchibouguac", "Hay Island", "Brion Island",
            "Mud Island", "Noddy Island", "Sable Island"),
  longitude = c(-61.690711993886, -62.99424426648467, -68.0330995267766, -68.86046523155126,
                -63.17714575179163, -64.86289734875632, -59.687854496181494, -61.49320012194406,
                -65.98883830489696, -65.98630641810254, -59.916718555363026),
  latitude = c(45.78560555624806, 49.52212432174339, 48.68187699980851, 48.40032409235994,
               45.835499821072546, 46.76743173640179, 46.033885657189906, 47.78602556317871,
               43.48307015235114, 43.46559859574744, 43.94331839714443))

sites_sf <- st_as_sf(sites, coords = c("longitude", "latitude"),
                     crs = 4326, agr = "constant")

## Generate empty tibble and list to store home range plots ------------------------------------------
df_hr_plot <- tibble(decade = character(),
                     period = character(),
                     plot = list())

# 1993-1999 ---------------------------------------------------------------------------------------------
decade_data <-filter(sf_hr, deploy.yr < 2000)
decade <- "1993-1999"

plot_hr <- ggplot(data = world) +
  geom_sf(fill= "darkgrey") +
  geom_sf(data = decade_data,
          aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
          alpha = 0.5,
          color = NA) +
  geom_sf(data = filter(sites_sf,sites %in% c("St George's Bay", "Anticosti Island",
                                              "Amet Island", "Kouchibouguac")),
          size = 2,
          shape = 18) +
  coord_sf(crs = st_crs(4326),
           xlim = c(bbox['xmin'], bbox['xmax']),
           ylim = c(bbox['ymin'], bbox['ymax'])) +
  labs(fill = "") +
  scale_fill_manual(values = period_fill, drop = T) +
  ggtitle(decade) +
  map_theme +
  theme(legend.position = "top")

# Store all individual parts to create plot
df_hr_plot <- bind_rows(df_hr_plot,
                        tibble(decade = decade,
                               period = "All",
                               plot = list(plot_hr)))

for (p in unique(decade_data$period)) {

# Generate home range plot for i individual in n month
plot_hr <-
  ggplot(data = world) +
  geom_sf(fill= "darkgrey") +
  geom_sf(data = filter(decade_data, period == p),
          aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
          alpha = 0.5,
          color = NA) +
  geom_sf(data = filter(sites_sf,sites %in% c("St George's Bay", "Anticosti Island",
                                              "Amet Island", "Kouchibouguac")),
          size = 2,
          shape = 18) +
  coord_sf(crs = st_crs(4326),
           xlim = c(bbox['xmin'], bbox['xmax']),
           ylim = c(bbox['ymin'], bbox['ymax'])) +
  labs(fill = "") +
  scale_fill_manual(values = period_fill, drop = T) +
  ggtitle(decade) +
  map_theme +
  theme(legend.position = "none")

# Store all individual parts to create plot
df_hr_plot <- bind_rows(df_hr_plot,
                        tibble(decade = decade,
                               period = p,
                               plot = list(plot_hr)))

}

# 2003-2008 -------------------------------------------------------------------------------------------
decade_data <-filter(sf_hr, between(deploy.yr, 2000, 2009))
decade <- "2003-2008"

plot_hr <- ggplot(data = world) +
  geom_sf(fill= "darkgrey") +
  geom_sf(data = decade_data,
          aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
          alpha = 0.5,
          color = NA) +
  geom_sf(data = filter(sites_sf,sites %in% c("Kouchibouguac")),
          size = 2,
          shape = 18) +
  coord_sf(crs = st_crs(4326),
           xlim = c(bbox['xmin'], bbox['xmax']),
           ylim = c(bbox['ymin'], bbox['ymax'])) +
  labs(fill = "") +
  scale_fill_manual(values = period_fill, drop = F) +
  ggtitle(decade) +
  map_theme +
  theme(legend.position = "top")

# Store all individual parts to create plot
df_hr_plot <- bind_rows(df_hr_plot,
                        tibble(decade = decade,
                               period = "All",
                               plot = list(plot_hr)))

for (p in unique(decade_data$period)) {

  # Generate home range plot for i individual in n month
  plot_hr <-
    ggplot(data = world) +
    geom_sf(fill= "darkgrey") +
    geom_sf(data = filter(decade_data, period == p),
            aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
            alpha = 0.5,
            color = NA) +
    geom_sf(data = filter(sites_sf,sites %in% c("Kouchibouguac")),
            size = 2,
            shape = 18) +
    coord_sf(crs = st_crs(4326),
             xlim = c(bbox['xmin'], bbox['xmax']),
             ylim = c(bbox['ymin'], bbox['ymax'])) +
    labs(fill = "") +
    scale_fill_manual(values = period_fill, drop = T) +
    ggtitle(decade) +
    map_theme +
    theme(legend.position = "none")

  # Store all individual parts to create plot
  df_hr_plot <- bind_rows(df_hr_plot,
                          tibble(decade = decade,
                                 period = p,
                                 plot = list(plot_hr)))

}

# 2013-2017 -------------------------------------------------------------------------------------------
decade_data <-filter(sf_hr, between(deploy.yr, 2010, 2017))
decade <- "2013-2017"

plot_hr <- ggplot(data = world) +
  geom_sf(fill= "darkgrey") +
  geom_sf(data = decade_data,
          aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
          alpha = 0.5,
          color = NA) +
  geom_sf(data = filter(sites_sf,sites %in% c("Kouchibouguac", "Brion Island")),
          size = 2,
          shape = 18) +
  coord_sf(crs = st_crs(4326),
           xlim = c(bbox['xmin'], bbox['xmax']),
           ylim = c(bbox['ymin'], bbox['ymax'])) +
  labs(fill = "") +
  scale_fill_manual(values = period_fill, drop = F) +
  ggtitle(decade) +
  map_theme +
  theme(legend.position = "top")

# Store all individual parts to create plot
df_hr_plot <- bind_rows(df_hr_plot,
                        tibble(decade = decade,
                               period = "All",
                               plot = list(plot_hr)))

for (p in unique(decade_data$period)) {

  # Generate home range plot for i individual in n month
  plot_hr <-
    ggplot(data = world) +
    geom_sf(fill= "darkgrey") +
    geom_sf(data = filter(decade_data, period == p),
            aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
            alpha = 0.5,
            color = NA) +
    geom_sf(data = filter(sites_sf,sites %in% c("Kouchibouguac", "Brion Island")),
            size = 2,
            shape = 18) +
    coord_sf(crs = st_crs(4326),
             xlim = c(bbox['xmin'], bbox['xmax']),
             ylim = c(bbox['ymin'], bbox['ymax'])) +
    labs(fill = "") +
    scale_fill_manual(values = period_fill, drop = T) +
    ggtitle(decade) +
    map_theme +
    theme(legend.position = "none")

  # Store all individual parts to create plot
  df_hr_plot <- bind_rows(df_hr_plot,
                          tibble(decade = decade,
                                 period = p,
                                 plot = list(plot_hr)))

}

# 2019-2022 ------------------------------------------------------------------------------------------
decade_data <-filter(sf_hr, between(deploy.yr, 2019, 2029))
decade <- "2019-2022"

plot_hr <- ggplot(data = world) +
  geom_sf(fill= "darkgrey") +
  geom_sf(data = decade_data,
          aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
          alpha = 0.5,
          color = NA) +
  geom_sf(data = filter(sites_sf,sites %in% c("Kouchibouguac", "Brion Island")),
          size = 2,
          shape = 18) +
  coord_sf(crs = st_crs(4326),
           xlim = c(bbox['xmin'], bbox['xmax']),
           ylim = c(bbox['ymin'], bbox['ymax'])) +
  labs(fill = "") +
  scale_fill_manual(values = period_fill, drop = F) +
  ggtitle(decade) +
  map_theme +
  theme(legend.position = "top")

# Store all individual parts to create plot
df_hr_plot <- bind_rows(df_hr_plot,
                        tibble(decade = decade,
                               period = "All",
                               plot = list(plot_hr)))

for (p in unique(decade_data$period)) {

  # Generate home range plot for i individual in n month
  plot_hr <-
    ggplot(data = world) +
    geom_sf(fill= "darkgrey") +
    geom_sf(data = filter(decade_data, period == p),
            aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
            alpha = 0.5,
            color = NA) +
    geom_sf(data = filter(sites_sf,sites %in% c("Kouchibouguac", "Brion Island")),
            size = 2,
            shape = 18) +
    coord_sf(crs = st_crs(4326),
             xlim = c(bbox['xmin'], bbox['xmax']),
             ylim = c(bbox['ymin'], bbox['ymax'])) +
    labs(fill = "") +
    scale_fill_manual(values = period_fill, drop = T) +
    ggtitle(decade) +
    map_theme +
    theme(legend.position = "none")

  # Store all individual parts to create plot
  df_hr_plot <- bind_rows(df_hr_plot,
                          tibble(decade = decade,
                                 period = p,
                                 plot = list(plot_hr)))

}
```

```
# Combined maps ---------------------------------------------------------------------------------------
# Legend
legend_gg <- ggpubr::get_legend(
  ggplot(data = sf_hr) +
    geom_sf(aes(fill = factor(period, levels = c("Summer", "Pre-breeding", "Breeding", "Post-breeding"))),
            alpha = 1,
            color = NA) +
    scale_fill_manual(values = period_fill, drop = T) +
    map_theme +
    labs(fill = "Period") +
    theme(legend.position = "top",
          legend.title.align=0.5,
          legend.box.margin = margin(t=-10, r=0, b=0, l=0)))

# All decades
hr_map_all <- plot_grid(
  plot_grid(
    plot_grid(
      ggpubr::text_grob("1993-1999", size = 10, vjust = 1.5),

      filter(df_hr_plot, decade == "1993-1999" & period == "Summer")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text.x = element_blank()),

      filter(df_hr_plot, decade == "1993-1999" & period == "Pre-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text.x = element_blank()),

      filter(df_hr_plot, decade == "1993-1999" & period == "Breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text.x = element_blank()),

      filter(df_hr_plot, decade == "1993-1999" & period == "Post-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text.x = element_blank()),

      ncol = 1, align = "vh", rel_heights = c(0.2, 1,1,1,1.1)),
    plot_grid(
      ggpubr::text_grob("2003-2008", size = 10, vjust = 1.5),

      filter(df_hr_plot, decade == "2003-2008" & period == "Summer")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2003-2008" & period == "Pre-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2003-2008" & period == "Breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2003-2008" & period == "Post-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      ncol = 1, align = "vh", rel_heights = c(0.2, 1,1,1,1.1)),

    plot_grid(
      ggpubr::text_grob("2013-2017", size = 10, vjust = 1.5),

      filter(df_hr_plot, decade == "2013-2017" & period == "Summer")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2013-2017" & period == "Pre-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2013-2017" & period == "Breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2013-2017" & period == "Post-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      ncol = 1, align = "vh", rel_heights = c(0.2, 1,1,1,1.1)),

    plot_grid(
      ggpubr::text_grob("2019-2022", size = 10, vjust = 1.5),

      filter(df_hr_plot, decade == "2019-2022" & period == "Summer")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2019-2022" & period == "Pre-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2019-2022" & period == "Breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      filter(df_hr_plot, decade == "2019-2022" & period == "Post-breeding")$plot[[1]] +
        theme(legend.position="none", text = element_text(size=8),
              plot.title = element_blank(), axis.text = element_blank()),

      ncol = 1, align = "vh", rel_heights = c(0.2, 1,1,1,1.1)),

    nrow = 1, rel_widths = c(4.57,4,4,4)),

  legend_gg,
  ncol = 1, rel_heights = c(1,0.05))

ggsave(filename = "Figures/Figure_A1.tiff",
       plot = hr_map_all,
       units = "cm",
       dpi = 300,
       height = 18.5,
       width = 16.5,
       bg = "white")
```

Tables

Table 1 - Deployments info

```
# Generate data -----------------------------------------------------------------------------------------------------

design <- hr.data %>%
  dplyr::select(id, deploy.yr, location, sexage) %>%
  mutate(location = case_when(location == "Brion Island" ~ "Center GSL",
                              TRUE ~ location)) %>%
  distinct(id, .keep_all = TRUE)

count_design_ageclass <- design %>%
  count(deploy.yr,location, sexage) %>%
  pivot_wider(names_from = sexage,
              values_from = n)

count_design_total <- design %>%
  count(deploy.yr,location, name = "Total")

count_design <- left_join(
  count_design_ageclass,
  count_design_total,
  by = c('deploy.yr', 'location')) %>%
  rename("Year" = deploy.yr, Location = location, 'Adult M' = 'adult M',
         'Adult F' = 'adult F', 'Juvenile F' = 'juvenile F', 'Juvenile M' = 'juvenile M')


# Generate Table ----------------------------------------------------------------------------------------------------

tab_design <- flextable((count_design)) %>%
  colformat_double(big.mark = "", digits = 0, na_str = "-") %>%
  add_header_row(top = TRUE,
                 values = c("Deployment", "", "","", "", ""),
                 colwidths = c(2,1,1,1,1,1)) %>%
  theme_booktabs(.) %>%
  align(align="center", part="all") %>%
  padding(padding=1.5, part="body") %>%
  padding(padding=0.5, part="header") %>%
  font(fontname="Time New Roman", part="all") %>%
  fontsize(size = 9, part = "all")

# Save table
save_as_docx(tab_design, path = "Figures/Table_1.docx")
```

Table 2 - Models results

```
# Generate data -----------------------------------------------------------------------------------------------------

# Extract parameters from model summary for area
mod_sum_area <- summary(d_yr_area)
mod_pars_area <- bind_rows(
  tibble_row(' ' = "Population-Level Effects:"),
  rownames_to_column(mod_sum_area[["fixed"]], " "),
  tibble_row(' ' = "Smooth Terms:"),
  rownames_to_column(mod_sum_area[["splines"]], " "),
  tibble_row(' ' = "Group-Level Effects:"),
  tibble_row(' ' = "~Id (Number of levels: 124)"),
  rownames_to_column(mod_sum_area[["random"]][["id"]], " ") %>%
    mutate(` ` = case_when(` ` == "sd(Intercept)" ~ "sd(Intercept_id)",
                           ` ` == "sd(sigma_Intercept)" ~ "sd(sigma_Intercept_id)",
                           ` ` == "cor(Intercept,sigma_Intercept)" ~ "cor(Intercept_id,sigma_Intercept_id)",
                           TRUE ~ ` `)),
  tibble_row(' ' = "~Year (Number of levels: 16)"),
  rownames_to_column(mod_sum_area[["random"]][["deploy.yr"]], " ") %>%
    mutate(` ` = case_when(` ` == "sd(Intercept)" ~ "sd(Intercept_yr)",
                           ` ` == "sd(sigma_Intercept)" ~ "sd(sigma_Intercept_yr)",
                           ` ` == "cor(Intercept,sigma_Intercept)" ~ "cor(Intercept_yr,sigma_Intercept_yr)",
                           TRUE ~ ` `))) %>%
  mutate(CI = if_else(is.na(Estimate),
                      NA_character_,
                      paste("(",format(round(`l-95% CI`,2),nsmall = 2), " - ",
                            format(round(`u-95% CI`,2),2), ")", sep = ""))) %>%
  dplyr::select(` `, Estimate, CI)

# Extract parameters from model summary for shape
mod_sum_shape <- summary(d_yr_shape)
```

```
mod_pars_shape <- bind_rows(
  tibble_row(' ' = "Population-Level Effects:"),
  rownames_to_column(mod_sum_shape[["fixed"]], " "),
  tibble_row(' ' = "Smooth Terms:"),
  rownames_to_column(mod_sum_shape[["splines"]], " "),
  tibble_row(' ' = "Group-Level Effects:"),
  tibble_row(' ' = "~Id (Number of levels: 124)"),
  rownames_to_column(mod_sum_shape[["random"]][["id"]], " ") %>%
    mutate(` ` = case_when(` ` == "sd(Intercept)" ~ "sd(Intercept_id)",
                           ` ` == "sd(sigma_Intercept)" ~ "sd(sigma_Intercept_id)",
                           ` ` == "cor(Intercept,sigma_Intercept)" ~ "cor(Intercept_id,sigma_Intercept_id)",
                           TRUE ~ ` `)),
  tibble_row(' ' = "~Year (Number of levels: 16)"),
  rownames_to_column(mod_sum_shape[["random"]][["deploy.yr"]], " ") %>%
    mutate(` ` = case_when(` ` == "sd(Intercept)" ~ "sd(Intercept_yr)",
                           ` ` == "sd(sigma_Intercept)" ~ "sd(sigma_Intercept_yr)",
                           ` ` == "cor(Intercept,sigma_Intercept)" ~ "cor(Intercept_yr,sigma_Intercept_yr)",
                           TRUE ~ ` `))) %>%
  mutate(CI = if_else(is.na(Estimate),
                      NA_character_,
                      paste("(",format(round(`l-95% CI`,2),nsmall = 2), " - ",
                            format(round(`u-95% CI`,2),2), ")", sep = ""))) %>%
  dplyr::select(` `, Estimate, CI)

# Extract parameters from model summary for centroid
mod_sum_cent <- summary(d_yr_cent)
mod_pars_cent <- bind_rows(
  tibble_row(' ' = "Population-Level Effects:"),
  rownames_to_column(mod_sum_cent[["fixed"]], " "),
  tibble_row(' ' = "Smooth Terms:"),
  rownames_to_column(mod_sum_cent[["splines"]], " "),
  tibble_row(' ' = "Group-Level Effects:"),
  tibble_row(' ' = "~Id (Number of levels: 124)"),
  rownames_to_column(mod_sum_cent[["random"]][["id"]], " ") %>%
    mutate(` ` = case_when(` ` == "sd(centx_Intercept)" ~ "sd(centx_Intercept_id)",
                           ` ` == "sd(sigma_centx_Intercept)" ~ "sd(sigma_centx_Intercept_id)",
                           ` ` == "cor(centx_Intercept,sigma_centx_Intercept)" ~ "cor(centx_Intercept_id,sigma_centx_Interce
                           ` ` == "sd(centy_Intercept)" ~ "sd(centy_Intercept_id)",
                           ` ` == "sd(sigma_centy_Intercept)" ~ "sd(sigma_centy_Intercept_id)",
                           ` ` == "cor(centy_Intercept,sigma_centy_Intercept)" ~ "cor(centy_Intercept_id,sigma_centy_Interce
                           TRUE ~ ` `)),
  tibble_row(' ' = "~Year (Number of levels: 16)"),
  rownames_to_column(mod_sum_cent[["random"]][["deploy.yr"]], " ") %>%
    mutate(` ` = case_when(` ` == "sd(centx_Intercept)" ~ "sd(centx_Intercept_yr)",
                           ` ` == "sd(sigma_centx_Intercept)" ~ "sd(sigma_centx_Intercept_yr)",
                           ` ` == "cor(centx_Intercept,sigma_centx_Intercept)" ~ "cor(centx_Intercept_yr,sigma_centx_Interce
                           ` ` == "sd(centy_Intercept)" ~ "sd(centy_Intercept_yr)",
                           ` ` == "sd(sigma_centy_Intercept)" ~ "sd(sigma_centy_Intercept_yr)",
                           ` ` == "cor(centy_Intercept,sigma_centy_Intercept)" ~ "cor(centy_Intercept_yr,sigma_centy_Interce
                           TRUE ~ ` `))) %>%
  dplyr::select(' ',Estimate, 'l-95% CI','u-95% CI')

# Filter parameters from centroid models to keep only cent.x parameters
mod_pars_centx <- mod_pars_cent %>%
  # Keep only cent x parameters
  filter(!(str_detect(` `, "centy"))) %>%
  mutate(
    # Remove centx_ character string
    ` ` = str_replace_all(` `, "centx_", ""),
    CI = if_else(is.na(Estimate),
                 NA_character_,
                 paste("(",format(round(`l-95% CI`,2),nsmall = 2), " - ",
                       format(round(`u-95% CI`,2),2), ")", sep = ""))) %>%
  dplyr::select(` `, Estimate, CI)

# Filter parameters from centroid models to keep only cent.y parameters
```

```r
mod_pars_centy <- mod_pars_cent %>%
  # Keep only cent y parameters
  filter(!(str_detect(` `, "centx"))) %>%
  mutate(
    # Remove centx_ character string
    ` ` = str_replace_all(` `, "centy_", ""),
    CI = if_else(is.na(Estimate),
                 NA_character_,
                 paste("(",format(round(`l-95% CI`,2),nsmall = 2), " - ",
                       format(round(`u-95% CI`,2),2), ")", sep = ""))) %>%
  dplyr::select(` `, Estimate, CI)

# Set desired parameters order for the table
pars <- c("Population-Level Effects:",
          "Intercept",
          "scalelogmean.loc.error",
          "scalesqrtn.loc",
          "locationNorthGSL",
          "locationSouthGSL",
          "periodMoult",
          "periodPost_breeding",
          "periodPre_breeding",
          "periodSummer","sexageadultM",
          "sexagejuvenileF",
          "sexagejuvenileM",
          "periodMoult:sexageadultM",
          "periodPost_breeding:sexageadultM",
          "periodPre_breeding:sexageadultM",
          "periodSummer:sexageadultM","periodMoult:sexagejuvenileF",
          "periodPost_breeding:sexagejuvenileF",
          "periodPre_breeding:sexagejuvenileF",
          "periodSummer:sexagejuvenileF","periodMoult:sexagejuvenileM",
          "periodPost_breeding:sexagejuvenileM",
          "periodPre_breeding:sexagejuvenileM",
          "periodSummer:sexagejuvenileM",
          "sigma_Intercept",
          "sscaledeploy.yr:periodBreeding_1",
          "sscaledeploy.yr:periodMoult_1",
          "sscaledeploy.yr:periodPost_breeding_1",
          "sscaledeploy.yr:periodPre_breeding_1",
          "sscaledeploy.yr:periodSummer_1",
          "Smooth Terms:",
          "sds(sscaledeploy.yrperiodBreeding_1)",
          "sds(sscaledeploy.yrperiodMoult_1)",
          "sds(sscaledeploy.yrperiodPost_breeding_1)",
          "sds(sscaledeploy.yrperiodPre_breeding_1)",
          "sds(sscaledeploy.yrperiodSummer_1)",
          "Group-Level Effects:",
          "~Id (Number of levels: 124)",
          "sd(Intercept_id)",
          "sd(sigma_Intercept_id)",
          "cor(Intercept_id,sigma_Intercept_id)",
          "~Year (Number of levels: 16)",
          "sd(Intercept_yr)",
          "sd(sigma_Intercept_yr)",
          "cor(Intercept_yr,sigma_Intercept_yr)")

# Merge parameters from all models
mod_pars <-
  full_join(mod_pars_area, mod_pars_shape, by=" ") %>%
  full_join(., mod_pars_centy, by=" ") %>%
  full_join(., mod_pars_centx, by=" ") %>%
  arrange(factor(` `, levels = pars)) %>%
  mutate(` ` = case_when(` ` == "scalelogmean.loc.error" ~ "Location error",
                         ` ` == "scalesqrtn.loc" ~ "Number of locations",
                         ` ` == "locationNorthGSL" ~ "North GSL",
                         ` ` == "locationSouthGSL" ~ "South GSL",
                         ` ` == "periodMoult" ~ "Moult",
                         ` ` == "periodPre_breeding" ~ "Pre-breeding",
                         ` ` == "periodPost_breeding" ~ "Post-breeding",
                         ` ` == "periodSummer" ~ "Summer",
                         ` ` == "sexageadultM" ~ "Adult M",
                         ` ` == "sexagejuvenileF" ~ "Juvenile F",
                         ` ` == "sexagejuvenileM" ~ "Juvenile M",
                         ` ` == "periodMoult:sexageadultM" ~ "Moult:Adult M",
                         ` ` == "periodPost_breeding:sexageadultM" ~ "Post-breeding:Adult M",
                         ` ` == "periodPre_breeding:sexageadultM" ~ "Pre-breeding:Adult M",
                         ` ` == "periodSummer:sexageadultM" ~ "Summer:Adult M",
                         ` ` == "periodMoult:sexagejuvenileM" ~ "Moult:Juvenile M",
                         ` ` == "periodPost_breeding:sexagejuvenileM" ~ "Post-breeding:Juvenile M",
                         ` ` == "periodPre_breeding:sexagejuvenileM" ~ "Pre-breeding:Juvenile M",
                         ` ` == "periodSummer:sexagejuvenileM" ~ "Summer:Juvenile M",
                         ` ` == "periodMoult:sexagejuvenileF" ~ "Moult:Juvenile F",
                         ` ` == "periodPost_breeding:sexagejuvenileF" ~ "Post-breeding:Juvenile F",
                         ` ` == "periodPre_breeding:sexagejuvenileF" ~ "Pre-breeding:Juvenile F",
                         ` ` == "periodSummer:sexagejuvenileF" ~ "Summer:Juvenile F",
                         ` ` == "sigma_Intercept" ~ "Intercept sigma",
                         ` ` == "sscaledeploy.yr:periodBreeding_1" ~ "Year:Breeding",
                         ` ` == "sscaledeploy.yr:periodMoult_1" ~ "Year:Moult",
                         ` ` == "sscaledeploy.yr:periodPost_breeding_1" ~ "Year:Post-breeding",
                         ` ` == "sscaledeploy.yr:periodPre_breeding_1" ~ "Year:Pre-breeding",
                         ` ` == "sscaledeploy.yr:periodSummer_1" ~ "Year:Summer",
                         ` ` == "sds(sscaledeploy.yrperiodBreeding_1)" ~ "Year(Breeding)",
                         ` ` == "sds(sscaledeploy.yrperiodMoult_1)" ~ "Year(Moult)",
                         ` ` == "sds(sscaledeploy.yrperiodPost_breeding_1)" ~ "Year(Post-breeding)",
                         ` ` == "sds(sscaledeploy.yrperiodPre_breeding_1)" ~ "Year(Pre-breeding)",
                         ` ` == "sds(sscaledeploy.yrperiodSummer_1)" ~ "Year(Summer)",
                         ` ` == "sd(Intercept_id)" ~ "sd(Intercept)",
                         ` ` == "sd(sigma_Intercept_id)" ~ "sd(sigma_Intercept)",
                         ` ` == "cor(Intercept_id,sigma_Intercept_id)" ~ "cor(Intercept,sigma_Intercept)",
                         ` ` == "sd(Intercept_yr)" ~ "sd(Intercept)",
                         ` ` == "sd(sigma_Intercept_yr)" ~ "sd(sigma_Intercept)",
                         ` ` == "cor(Intercept_yr,sigma_Intercept_yr)" ~ "cor(Intercept,sigma_Intercept)",
                         TRUE ~ ` `))

# Generate table ----------------------------------------------------------------------------------------------------

# Generate table
tab_pars <- flextable((mod_pars)) %>%
  colformat_double(big.mark = ",", digits = 2, na_str = "-") %>%
  merge_h_range(i = c(1,27,32,33,37), j1 = " ", j2 = "CI.y.y") %>%
  merge_h_range(i = c(1), j1 = "Estimate.x", j2 = "CI.x", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.y", j2 = "CI.y", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.x.x", j2 = "CI.x.x", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.y.y", j2 = "CI.y.y", part = "header") %>%
  compose(part = "header",
          j = c(2:9),
          value = c(as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"))) %>%
  add_header_row(top = TRUE,
                 values = c("","", "","Latitude", "Longitude"),
                 colwidths = c(1,2,2,2,2)) %>%
  add_header_row(top = TRUE,
                 values = c("","Area", "Shape", "Centroid"),
                 colwidths = c(1,2,2,4)) %>%
  theme_booktabs(.) %>%
  hline(i=c(1,27,32)) %>%
  align(align="left", j=c(1), part="all") %>%
  align(align="left", j=c(3,5,7,9), part="body") %>%
  align(align="right", j=c(2,4,6,8), part="body") %>%
  align(align="center", i = c(1,2,3), part="header") %>%
  padding(padding=1.5, part="body") %>%
  padding(padding=1, part="header") %>%
  padding(padding=1,i = c(1,27,32,33,37)) %>%
  font(fontname="Time New Roman", part="all") %>%
  fontsize(size = 9, part = "all") %>%
  bold(i = c(1,27,32,33,37)) %>%
  height(i=c(1,2,3), height = 0.4, unit = "cm", part = "header") %>%
  width(j=c(1), width = 4.4, unit = "cm") %>%
  width(j=c(2,4), width = 0.8, unit = "cm") %>%
  width(j=c(3,5), width = 2.1, unit = "cm") %>%
  width(j=c(6), width = 0.9, unit = "cm") %>%
  width(j=c(7), width = 2.2, unit = "cm") %>%
  width(j=c(8), width = 1, unit = "cm") %>%
  width(j=c(9), width = 2.4, unit = "cm") %>%
  set_table_properties(layout = "fixed")

# Save table
save_as_docx(tab_pars, path = "Figures/Table_2.docx")
```

Table 3 - Interindividual differences

```
# Generate data -----------------------------------------------------------------------------------------------------

# Area
pers_pred_results_area <-
  data.frame(model = c("Mean Model", "", "", "Dispersion Model", ""),
             type = c("Rpm", "Rpm_adj", "CVm", "Rp_var", "CV_var"),
             Est = c(mean(Rp_mu_area), mean(Rp_mu_adj_area), mean(CV_mu_area), mean(Rp_var_area), mean(CV_var_area)),
             L.CI= c(quantile(Rp_mu_area,0.025), quantile(Rp_mu_adj_area,0.025), quantile(CV_mu_area,0.025),
                     quantile(Rp_var_area,0.025),quantile(CV_var_area,0.025)),
             U.CI= c(quantile(Rp_mu_area,0.975), quantile(Rp_mu_adj_area,0.975), quantile(CV_mu_area,0.975),
                     quantile(Rp_var_area,0.975),quantile(CV_var_area,0.975))) %>%
  mutate(Est = format(round(`Est`,3),3),
         CI = paste("(",format(round(`L.CI`,3),nsmall = 3), " - ",
                    format(round(`U.CI`,3),3), ")", sep = "")) %>%
  dplyr::select(type, Est, CI)


pers_pred_area <- pers_pred_results_area %>%
  dplyr::select(type, Est) %>%
  add_row(!!! setNames(list("Mean Model", NA_character_),
                       names(.)), .before = 1) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_area$CI[1]),
                       names(.)), .after = 2) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_area$CI[2]),
                       names(.)), .after = 4) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_area$CI[3]),
                       names(.)), .after = 6) %>%
  add_row(!!! setNames(list("Dispersion Model", NA_character_),
                       names(.)), .before = 8) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_area$CI[4]),
                       names(.)), .after = 9) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_area$CI[5]),
                       names(.)), .after = 11)

# Shape
pers_pred_results_shape <-
  data.frame(model = c("Mean Model", "", "", "Dispersion Model", ""),
             type = c("Rpm", "Rpm_adj", "CVm", "Rp_var", "CV_var"),
             Est = c(mean(Rp_mu_shape), mean(Rp_mu_adj_shape), mean(CV_mu_shape),
                     mean(Rp_var_shape), mean(CV_var_shape)),
             L.CI= c(quantile(Rp_mu_shape,0.025), quantile(Rp_mu_adj_shape,0.025), quantile(CV_mu_shape,0.025),
                     quantile(Rp_var_shape,0.025),quantile(CV_var_shape,0.025)),
             U.CI= c(quantile(Rp_mu_shape,0.975), quantile(Rp_mu_adj_shape,0.975), quantile(CV_mu_shape,0.975),
                     quantile(Rp_var_shape,0.975),quantile(CV_var_shape,0.975))) %>%
  mutate(Est = format(round(`Est`,3),3),
         CI = paste("(",format(round(`L.CI`,3),nsmall = 3), " - ",
                    format(round(`U.CI`,3),3), ")", sep = "")) %>%
  dplyr::select(type, Est, CI)


pers_pred_shape <- pers_pred_results_shape %>%
  dplyr::select(type, Est) %>%
  add_row(!!! setNames(list("Mean Model", NA_character_),
                       names(.)), .before = 1) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_shape$CI[1]),
                       names(.)), .after = 2) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_shape$CI[2]),
                       names(.)), .after = 4) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_shape$CI[3]),
                       names(.)), .after = 6) %>%
  add_row(!!! setNames(list("Dispersion Model", NA_character_),
                       names(.)), .before = 8) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_shape$CI[4]),
                       names(.)), .after = 9) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_shape$CI[5]),
                       names(.)), .after = 11)

# Latitude (cent.y)
pers_pred_results_centy <-
  data.frame(model = c("Mean Model", "", "", "Dispersion Model", ""),
             type = c("Rpm", "Rpm_adj", "CVm", "Rp_var", "CV_var"),
             Est = c(mean(Rp_mu_centy), mean(Rp_mu_adj_centy), mean(CV_mu_centy),
                     mean(Rp_var_centy), mean(CV_var_centy)),
             L.CI= c(quantile(Rp_mu_centy,0.025), quantile(Rp_mu_adj_centy,0.025), quantile(CV_mu_centy,0.025),
                     quantile(Rp_var_centy,0.025),quantile(CV_var_centy,0.025)),
             U.CI= c(quantile(Rp_mu_centy,0.975), quantile(Rp_mu_adj_centy,0.975), quantile(CV_mu_centy,0.975),
                     quantile(Rp_var_centy,0.975),quantile(CV_var_centy,0.975))) %>%
  mutate(Est = format(round(`Est`,3),3),
         CI = paste("(",format(round(`L.CI`,3),nsmall = 3), " - ",
                    format(round(`U.CI`,3),3), ")", sep = "")) %>%
  dplyr::select(type, Est, CI)

pers_pred_centy <- pers_pred_results_centy %>%
  dplyr::select(type, Est) %>%
  add_row(!!! setNames(list("Mean Model", NA_character_),
                       names(.)), .before = 1) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centy$CI[1]),
                       names(.)), .after = 2) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centy$CI[2]),
                       names(.)), .after = 4) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centy$CI[3]),
                       names(.)), .after = 6) %>%
  add_row(!!! setNames(list("Dispersion Model", NA_character_),
                       names(.)), .before = 8) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centy$CI[4]),
                       names(.)), .after = 9) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centy$CI[5]),
                       names(.)), .after = 11)

# Longitude (cent.x)
pers_pred_results_centx <-
  data.frame(model = c("Mean Model", "", "", "Dispersion Model", ""),
             type = c("Rpm", "Rpm_adj", "CVm", "Rp_var", "CV_var"),
             Est = c(mean(Rp_mu_centx), mean(Rp_mu_adj_centx), mean(CV_mu_centx),
                     mean(Rp_var_centx), mean(CV_var_centx)),
             L.CI= c(quantile(Rp_mu_centx,0.025), quantile(Rp_mu_adj_centx,0.025), quantile(CV_mu_centx,0.025),
                     quantile(Rp_var_centx,0.025),quantile(CV_var_centx,0.025)),
             U.CI= c(quantile(Rp_mu_centx,0.975), quantile(Rp_mu_adj_centx,0.975), quantile(CV_mu_centx,0.975),
                     quantile(Rp_var_centx,0.975),quantile(CV_var_centx,0.975))) %>%
  mutate(Est = format(round(`Est`,3),3),
         CI = paste("(",format(round(`L.CI`,3),nsmall = 3), " - ",
                            format(round(`U.CI`,3),3), ")", sep = "")) %>%
  dplyr::select(type, Est, CI)

pers_pred_centx <- pers_pred_results_centx %>%
  dplyr::select(type, Est) %>%
  add_row(!!! setNames(list("Mean Model", NA_character_),
                       names(.)), .before = 1) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centx$CI[1]),
                       names(.)), .after = 2) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centx$CI[2]),
                       names(.)), .after = 4) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centx$CI[3]),
                       names(.)), .after = 6) %>%
  add_row(!!! setNames(list("Dispersion Model", NA_character_),
                       names(.)), .before = 8) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centx$CI[4]),
                       names(.)), .after = 9) %>%
  add_row(!!! setNames(list(NA_character_, pers_pred_results_centx$CI[5]),
                       names(.)), .after = 11)

# Combine all home range characteristics
pers_pred_results <-
  bind_cols(pers_pred_area,
            pers_pred_shape[,-1],
            pers_pred_centy[,-1],
            pers_pred_centx[,-1])

# Generate table ---------------------------------------------------------------------------------------------

tab_perso <- flextable::flextable(pers_pred_results) %>%
  colformat_double(big.mark = ",", digits = 3, na_str = "-") %>%
  merge_h_range(i = c(1,8), j1 = "type", j2 = "...5") %>%
  compose(part = "header",
          j = c(1:5),
          value = c(as_paragraph("Estimate Type"),
                    as_paragraph("Mean Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph("Mean Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph("Mean Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph("Mean Estimate (CI", as_sub("95%"), ")"))) %>%
  add_header_row(top = TRUE,
                 values = c("","", "", "Latitude", "Longitude"),
                 colwidths = c(1,1,1,1,1)) %>%
  add_header_row(top = TRUE,
                 values = c("","Area", "Shape", "Centroid"),
                 colwidths = c(1,1,1,2)) %>%
  compose(part = "body",
          i = c(1:12),
          j = 1,
          value = c(as_paragraph("Mean Model"),
                    as_paragraph("Rp", as_sub("m")),
                    as_paragraph("Rp", as_sub("m")),
                    as_paragraph("Rp adj", as_sub("m")),
                    as_paragraph("Rp adj", as_sub("m")),
                    as_paragraph("CV", as_sub("m")),
                    as_paragraph("CV", as_sub("m")),
                    as_paragraph("Dispersion Model"),
                    as_paragraph("Rp", as_sub("v")),
                    as_paragraph("Rp", as_sub("v")),
                    as_paragraph("CV", as_sub("v")),
                    as_paragraph("CV", as_sub("v")))) %>%
  merge_v(.) %>%
  bold(i = c(1,8)) %>%
```

```r
  theme_booktabs(.) %>%
  hline(i=c(1,8)) %>%
  align(align="center", j=c(1), i=c(2:7,9:12), part="body") %>%
  align(align="center", j=c(2:5), part="body") %>%
  align(align="center", i = c(1,2,3), part="header") %>%
  align(align="center", i = c(3), j=c(1), part="header") %>%
  padding(padding=1.5, part="body") %>%
  padding(padding=1, part="header") %>%
  padding(padding=1, i = c(1,8)) %>%
  padding(padding.bottom=0, i = c(2,4,6,9,11)) %>%
  padding(padding.top=0, i = c(3,5,7,10,12)) %>%
  padding(padding.bottom=5, i = c(3,5,7,10)) %>%
  font(fontname="Times New Roman", part="all") %>%
  fontsize(size = 9, part = "all") %>%
  width(j=c(1), width = 1.5, unit = "cm") %>%
  width(j=c(2:5), width = 2.4, unit = "cm") %>%
  set_table_properties(layout = "fixed")

# Save table
save_as_docx(tab_perso, path = "Figures/Table_3.docx")
```

Table S1 - Post-hoc comparison between periods, within sexage classes

```r
# Generate data -----------------------------------------------------------------------------------------------------

# Post-hoc tables combined
h_all_comb_seas <- bind_rows(
  tibble_row('Hypothesis' = "Area"),
  h_all_area_seas,
  tibble_row('Hypothesis' = "Shape"),
  h_all_shape_seas,
  tibble_row('Hypothesis' = "Centroid Latitude"),
  h_all_centy_seas,
  tibble_row('Hypothesis' = "Centroid Longitude"),
  h_all_centx_seas)

# Generate table ----------------------------------------------------------------------------------------------------
h_tab_comb_seas <-
  flextable((h_all_comb_seas)) %>%
  colformat_double(big.mark = ",", digits = 2, na_str = "-") %>%
  merge_h_range(i = c(1,8,15,22), j1 = "Hypothesis", j2 = "Star.y.y") %>%
  merge_h_range(i = c(1), j1 = "Estimate.x", j2 = "CI.x", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.y", j2 = "CI.y", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.x.x", j2 = "CI.x.x", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.y.y", j2 = "CI.y.y", part = "header") %>%
  compose(part = "header",
          j = c(1:21),
          i = c(1),
          value = c(as_paragraph(""),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"))) %>%
  add_header_row(top = TRUE,
                 values = c("", "Adult F", "Adult M",
                            "Juvenile F", "Juvenile M"),
                 colwidths = c(1,5,5,5,5)) %>%
  theme_booktabs(.) %>%
  hline(i=c(1,8,15,22)) %>%
  align(align="center", part="all") %>%
  align(align="left", j=c(1), part="all") %>%
  align(align="left", j=c(3,8,13,18), part="body") %>%
  align(align="right", j=c(2,7,12,17), part="body") %>%
  align(align="center", i = c(1,2), part="header") %>%
  padding(padding=1.5, part="body") %>%
  padding(padding=0.5, part="header") %>%
  font(fontname="Time New Roman", part="all") %>%
  fontsize(size = 8, part = "all") %>%
  bold(i = c(1,8,15,22)) %>%
  width(j=c(1), width = 1.9, unit = "cm") %>%
  width(j=c(2,7,12,17), width = 0.7, unit = "cm") %>%
  width(j=c(3,8,13,18), width = 1.8, unit = "cm") %>%
  width(j=c(4,9,14,19), width = 1.2, unit = "cm") %>%
  width(j=c(5,10,15,20), width = 1.5, unit = "cm") %>%
  width(j=c(6,11,16,21), width = 0.6, unit = "cm") %>%
  set_table_properties(layout = "fixed")

# Save table
save_as_docx(h_tab_comb_seas,
             path = "Figures/Table_S1.docx",
             pr_section = prop_section(
               page_size = page_size(orient = "landscape")))
```

Table S2 - Post-hoc comparison between sexage classes, within periods

```
# Generate data -----------------------------------------------------------------------------------------------------

# Post-hoc tables combined
h_all_comb_onto <- bind_rows(
  tibble_row('Hypothesis' = "Area"),
  h_all_area_onto,
  tibble_row('Hypothesis' = "Shape"),
  h_all_shape_onto,
  tibble_row('Hypothesis' = "Centroid Latitude"),
  h_all_centy_onto,
  tibble_row('Hypothesis' = "Centroid Longitude"),
  h_all_centx_onto)

# Generate table ----------------------------------------------------------------------------------------------------
h_tab_comb_onto <-
  flextable((h_all_comb_onto)) %>%
  colformat_double(big.mark = ",", digits = 2, na_str = "-") %>%
  merge_h_range(i = c(1,8,15,22), j1 = "Hypothesis", j2 = "Star.y.y") %>%
  merge_h_range(i = c(1), j1 = "Estimate.x", j2 = "CI.x", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.y", j2 = "CI.y", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.x.x", j2 = "CI.x.x", part = "header") %>%
  merge_h_range(i = c(1), j1 = "Estimate.y.y", j2 = "CI.y.y", part = "header") %>%
  compose(part = "header",
          j = c(1:21),
          i = c(1),
          value = c(as_paragraph(""),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"),
                    as_paragraph("Estimate (CI", as_sub("95%"), ")"),
                    as_paragraph(""),
                    as_paragraph("Evidence Ratio"),
                    as_paragraph("Posterior Probability"),
                    as_paragraph("Star"))) %>%
  add_header_row(top = TRUE,
                 values = c("", "Summer", "Pre-breeding",
                            "Breeding", "Post-breeding"),
                 colwidths = c(1,5,5,5,5)) %>%
  theme_booktabs(.) %>%
  hline(i=c(1,8,15,22)) %>%
  align(align="center", part="all") %>%
  align(align="left", j=c(1), part="all") %>%
  align(align="left", j=c(3,8,13,18), part="body") %>%
  align(align="right", j=c(2,7,12,17), part="body") %>%
  align(align="center", i = c(1,2), part="header") %>%
  padding(padding=1.5, part="body") %>%
  padding(padding=0.5, part="header") %>%
  font(fontname="Time New Roman", part="all") %>%
  fontsize(size = 8, part = "all") %>%
  bold(i = c(1,8,15,22)) %>%
  width(j=c(1), width = 1.6, unit = "cm") %>%
  width(j=c(2,7,12,17), width = 0.7, unit = "cm") %>%
  width(j=c(3,8,13,18), width = 1.8, unit = "cm") %>%
  width(j=c(4,9,14,19), width = 1.2, unit = "cm") %>%
  width(j=c(5,10,15,20), width = 1.5, unit = "cm") %>%
  width(j=c(6,11,16,21), width = 0.6, unit = "cm") %>%
  set_table_properties(layout = "fixed")

# Save table
save_as_docx(h_tab_comb_onto,
             path = "Figures/Table_S2.docx",
             pr_section = prop_section(
               page_size = page_size(orient = "landscape")))
```