

R-code for ‘Effect individual and maternal inbreeding on pup early development’

Compiled by A.J. Paijmans and A.L. Berthelsen

19 October, 2023

Contents

| | |
|---|-----------|
| Packages and libraries | 2 |
| Microsatellite dataset | 2 |
| SNP dataset | 4 |
| Variance in inbreeding | 5 |
| Statistical models - microsatellite heterozygosity | 7 |
| Microsatellite dataset: pup birth mass | 7 |
| Model 1: including maternal effects | 7 |
| Residual check of model | 8 |
| Model 2: excluding maternal effects | 9 |
| Residual check of model | 10 |
| Parameter estimates pup birth mass models | 11 |
| Microsatellite dataset: survival | 12 |
| Model 1: including maternal effects | 12 |
| Residual check of model | 13 |
| Model 2: excluding maternal effects | 14 |
| Residual check of model | 15 |
| Parameter estimates pup survival models | 16 |
| Microsatellite dataset: pup weight gain | 17 |
| Model 1: including maternal effects | 17 |
| Residual check of model | 18 |
| Model 2: excluding maternal effects | 19 |
| Residual check of model | 20 |
| Parameter estimates pup survival models | 21 |
| Statistical models - SNP inbreeding | 23 |
| Birth weight analysis | 23 |
| Survival analysis | 26 |
| Growth curves with repeated measures | 30 |
| Repeated measures growth curve models | 32 |
| Individual growth curves | 43 |
| Manuscript figures | 46 |
| Figure 1: map and seasonal data | 46 |
| Figure 2: forest plots microsatellite models | 51 |
| Figure 3: forest plots SNP models | 55 |

This document contains all the R code used in the workflow for the manuscript *No evidence for inbreeding depression for birth mass, growth and survival in Antarctic fur seal pups* by Anneke J. Paijmans, Ane Liv Bertholdson, Rebecca Nagel, Felicitas Cristaller, Nicole Kröcker, Jaume Forcada, and Joseph I. Hoffman. The R Markdown file and the raw data can all be downloaded via Zenodo, [LINK](#). Additional scripts are available on Github, [LINK](#). Please, don't hesitate to contact me if you have any questions: a.paijmans[at]uni-bielefeld.de.

Packages and libraries

[CHECK LATER IF ALL ARE NECESSARY/INCLUDED!]

```
packages <- c("here",
             "readxl",
             "tidyverse",
             "inbreedR",
             "lme4",
             "lmerTest",
             "DHARMa",
             "sjPlot",
             "cowplot",
             "ggtext",
             "nlme",
             "car",
             "AICcmodavg",
             "data.table",
             "rcartocolor",
             "dotwhisker",
             "merDeriv")

# # Install packages needed not yet installed
# installed_packages <- packages %in% rownames(installed.packages())
# if (any(installed_packages == FALSE)) {
#   install.packages(packages[!installed_packages])
# }

# Load packages
invisible(lapply(packages, library, character.only = TRUE))
```

Microsatellite dataset

Before starting the analysis, mother-pup pairs were checked for genetic matching using the excel NEWPAT macro (REF). The genotypes of pairs with a maximum of 3 mismatching loci were visually inspected. If a scoring mistake was identified, the genotype was corrected. The maternity analysis was then rerun on the updated microsatellite data. Mother-pup pairs with fewer than 2 mismatching loci were considered a genetic match.

Preparing the files for the maternity analysis and postprocessing of the results were done in separate R scripts. These scripts and the NEWPAT excel macro files can be found on github [LINK](#).

First, we load the fitness data including the results of the maternity analysis, and the microsatellite data for all our individuals in two separate dataframes.

```
# Load pup data
pup_data <- openxlsx::read.xlsx(here("Data", "Processed",
  ↪ "pup_growth_mums_checked.xlsx"), detectDates = T) %>%
  mutate(Pup_Sex = as.factor(Pup_Sex)) %>%
  mutate(Year = as.factor(Year)) %>%
  mutate(Age_Tag = as.integer(Age_Tag))
```

```
# Load msats for all individuals in this study
msats_all <- openxlsx::read.xlsx(here("Data", "Processed",
  ↪ "msats_growth_individuals.xlsx"), detectDates = T) %>%
  mutate(Gaps = as.numeric(Gaps))
```

Genotypes with more than 4 missing loci out of 39 were removed.

```
# Keep only individuals genotyped for all 39 loci, with a maximum of 4 missing loci
msats_all <- msats_all %>%
  filter(Gaps < 5)
```

We then calculated sMLH for all remaining individuals and added this to the fitness data.

```
# Convert msat data to inbreedR input
msat_genotypes39 <- msats_all %>% select(Pv9.a:Mang36.b)
msat_ids <- msats_all %>% select(uniqueID)

msat_genotypes39_raw <- convert_raw(msat_genotypes39)

# Check if data is in right format for InbreedR
check_data(msat_genotypes39_raw, num_ind = nrow(msat_ids), num_loci =
  ↪ length(msat_genotypes39_raw))
```

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

```
# Calculate sMLH (incl FWB)
het39 <- sMLH(msat_genotypes39_raw)

sMLH_msat39 <- cbind(msat_ids, het39)
colnames(sMLH_msat39) <- c("uniqueID", "sMLH_msat39")

# Add sMLH to pup data
pup_data <- left_join(pup_data, sMLH_msat39, by = c("uniqueID_pup" = "uniqueID")) %>%
  rename(sMLH_msat39_pup=sMLH_msat39)

pup_data <- left_join(pup_data, sMLH_msat39, by = c("uniqueID_mum" = "uniqueID")) %>%
  rename(sMLH_msat39_mum=sMLH_msat39)
```

We then removed all mothers and maternal information for the mothers that were not a genetic match (ie more than 1 mismatching locus).

```
#~~ Remove mums that are not a genetic match, but keep pups (we can still use them in the
  ↪ analysis without mums)
# This means we also have to remove mum birthyear and sMLH for the mismatching mums
pup_data <- pup_data %>%
  mutate(ID_Mum = ifelse(gen_mum == "match" | is.na(gen_mum), ID_Mum, NA)) %>%
  mutate(uniqueID_mum = ifelse(gen_mum == "mismatch", NA, uniqueID_mum)) %>%
```

```
mutate(MumBirthYear = ifelse(gen_mum == "mismatch", NA, MumBirthYear)) %>%
mutate(Mum_Age = ifelse(gen_mum == "mismatch", NA, Mum_Age)) %>%
mutate(sMLH_msat39_mum = ifelse(gen_mum == "mismatch", NA, sMLH_msat39_mum)) %>%
select(-c(gen_mum, MISMATCHES))
```

And finally added a column with survival information (1 = survived until tagging, 0 = died).

```
pup_data <- pup_data %>%
  mutate(Survival = ifelse(!is.na(Pup_Death), "0",
                           ifelse(!is.na(Pup_TagWeight) & !is.na(Pup_Death), "0",
                                   ifelse(!is.na(Pup_TagWeight) & is.na(Pup_Death), "1",
                                         NA)))) %>%
  mutate(Survival = as.factor(Survival))

# This leaves all pups that do not have a tagging weight and also no death date with NA
#   for Survival.
# These will be removed: we wont be able to use them in the growth model nor the survival
#   model (as we do not know whether they were dead or alive).
# So for consistency, we will also not use them in the birth mass model (n=179)
# All pups with a 2nd weight and no death data are assumed to have survived at least
#   until the end of the season

pup_data <- pup_data %>%
  filter(!is.na(Survival))
```

Making sure all variables have the right categories

```
# Fix column categories
pup_data <- pup_data %>%
  mutate(Pup_Sex = as.factor(Pup_Sex)) %>%
  mutate(Year = as.factor(Year)) %>%
  mutate(Age_Tag = as.integer(Age_Tag)) %>%
  mutate(Survival = as.factor(Survival))
```

SNP dataset

Datasets for birth weight, survival and repeated measures growth analysis. Scripts for SNP data filtering, calculating sMLH and F_{ROH} are available on github [LINK](#).

```
DataRM_Day60 <- read.csv(here("Data", "Raw", "GrowthRM_BI1820_Day60.new.csv"))

Unique_Day60 <- subset(DataRM_Day60, DataRM_Day60$Age_Days < 62 &
                       !DataRM_Day60$ID == 'H2' &
                       !DataRM_Day60$ID == 'H5') %>%
  mutate(Sex = as.factor(Sex)) %>%
  mutate(Season = as.factor(Season)) %>%
  mutate(Beach = as.factor(Beach)) %>%
  distinct(ID, .keep_all = TRUE)
#98 pups

UniqueSurvivors_Day60 <- subset(Unique_Day60, Unique_Day60$Death == 'N')
#76 pups

SurvivorsRM_Day60 <- subset(DataRM_Day60, DataRM_Day60$Death == 'N' &
```

```

DataRM_Day60$Age_Days < 62 &
!DataRM_Day60$ID == 'H2' &
!DataRM_Day60$ID == 'H5') %>%
mutate(Sex = as.factor(Sex)) %>%
mutate(Season = as.factor(Season)) %>%
mutate(Beach = as.factor(Beach)) %>%
mutate(Age_Days = as.numeric(Age_Days))

```

Variance in inbreeding

To explore the variance in inbreeding, we also calculated g_2 (n permutations = 1000, n bootstraps = 1000). Since the g_2 calculations can take some time, we calculated it once, stored it as a Gdata object and load the data from the saved object. The code to calculate the g_2 can be found on Github [LINK](#), as well as the resulting Rdata objects.

```
load(here("Data", "Processed", "g2_39loci_after_checks.RData"))
```

```
# summary
g2_39loci
```

```
##
##
## Calculation of identity disequilibrium with g2 for microsatellite data
## -----
##
## Data: 1490 observations at 39 markers
## Function call = g2_microsats(genotypes = msat_genotypes39_raw, nperm = 1000,      nboot = 1000, CI = 0.95)
##
## g2 = 0.0005543019, se = 0.0003344484
##
## confidence interval
##      2.5%      97.5%
## -6.79188e-05  1.25651e-03
##
## p (g2 > 0) = 0.041 (based on 1000 permutations)
```

```
load(here("Data", "Processed", "g2_snps.RData"))
```

```
# summary SNPs
g2_snp_geno
```

```
##
##
## Calculation of identity disequilibrium with g2 for SNP data
## -----
##
## Data: 196 observations at 77417 markers
## Function call = g2_snps(genotypes = snp_genotypes, nperm = 1000, nboot = 1000,      CI = 0.95, parallel = FALSE)
##
## g2 = 0.000163467, se = 2.081776e-05
##
## confidence interval
##      2.5%      97.5%
```

```

## 0.0001242281 0.0002068790
##
## p (g2 > 0) = 0.001 (based on 999 permutations)

# Make df for plotting
g2_plot <- rbind(data.frame(g2_boot = g2_snp_geno$g2_boot, gen_data = "snps"),
                 data.frame(g2_boot = g2_39loci$g2_boot, gen_data = "msats"))

lcl_snps <- g2_snp_geno$CI_boot[1]
ucl_snps <- g2_snp_geno$CI_boot[2]
g2_boot_summary_snps <- data.frame(lcl_snps, ucl_snps)

lcl_msat <- g2_39loci$CI_boot[1]
ucl_msat <- g2_39loci$CI_boot[2]
g2_boot_summary_msat <- data.frame(lcl_msat, ucl_msat)

# Colors
col1 <- "#872ca2"
col2 <- "#f6a97a"

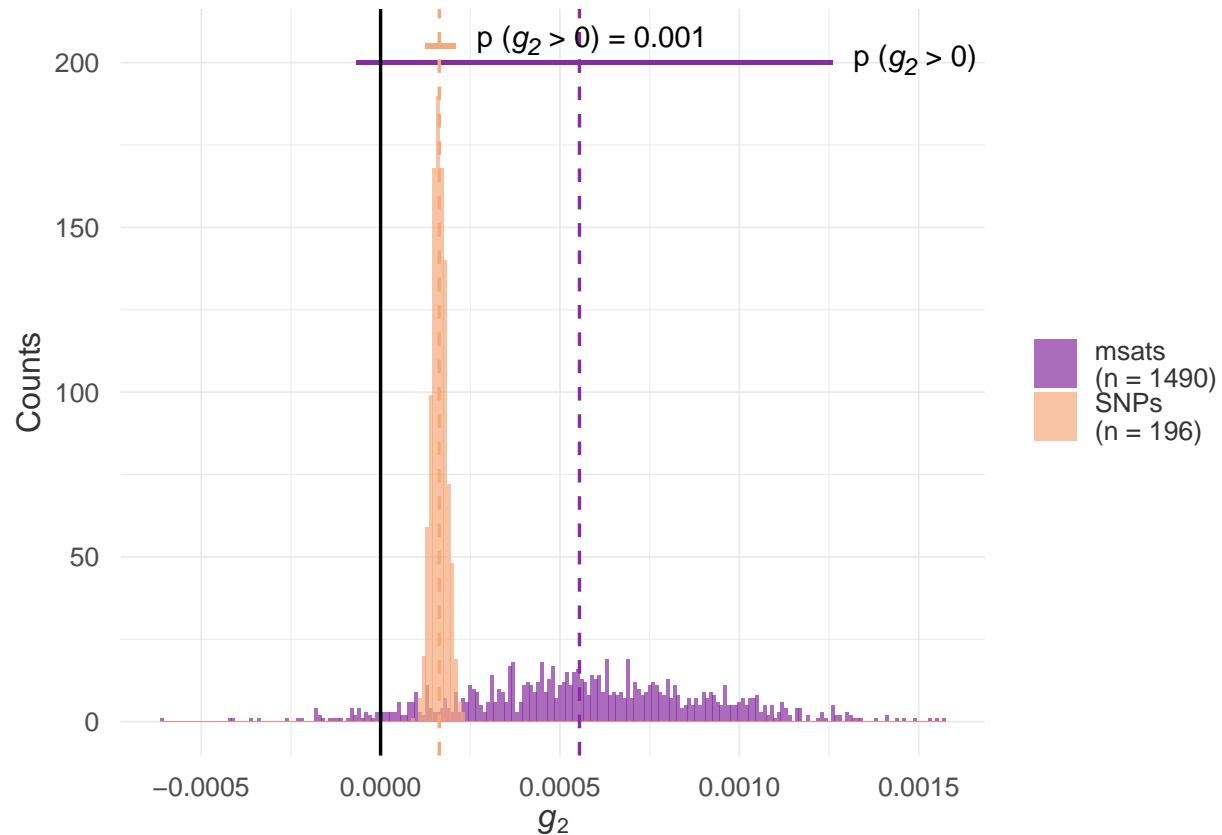
# P values
p_msats <- paste0("p (*g<sub>2</sub>* > 0) = ", g2_39loci$p_val)
p_snps <- paste0("p (*g<sub>2</sub>* > 0) = ", g2_snp_geno$p_val)

# Use Martin Stoffel's GGplot theme as a base
source(here("Rcode", "anneke_theme.R"))

ggplot(g2_plot, aes(x = g2_boot, fill = gen_data)) +
  geom_histogram(alpha = 0.7, position = "identity", binwidth = 0.00001) + # 0.00001 or
  ↪ 0.00005 ?
  scale_fill_manual(values = c(col1, col2), labels = c("msats\n(n = 1490)", "SNPs\n(n =
  ↪ 196)")) +
  # Add CI bars and g2 line for msats
  geom_errorbarh(aes(xmin = g2_boot_summary_msat$lcl_msat , xmax =
  ↪ g2_boot_summary_msat$ucl_msat , y = 200),
                 linewidth = 0.8, color = col1, linetype = "solid", height = 0) +
  geom_vline(xintercept = g2_39loci$g2, linewidth = 0.6, color = col1, linetype =
  ↪ "dashed") +
  # Add CI bars and g2 line for SNPs
  geom_errorbarh(aes(xmin = g2_boot_summary_snps$lcl_snps , xmax =
  ↪ g2_boot_summary_snps$ucl_snps , y = 205),
                 linewidth = 0.8, color = col2, linetype = "solid", height = 0) +
  geom_vline(xintercept = g2_snp_geno$g2, linewidth = 0.6, color = col2, linetype =
  ↪ "dashed") +
  # Add zero line
  geom_vline(xintercept = 0, linewidth = 0.6, linetype = "solid") +
  # Add p values
  annotate("richtext", x=g2_boot_summary_msat$ucl_msat, y=200, label = p_msats, fill =
  ↪ NA, label.color = NA, hjust = -0.05, size = 4) +
  annotate("richtext", x=g2_boot_summary_snps$ucl_snps, y=206, label=p_snps, fill = NA,
  ↪ label.color = NA, hjust = -0.05, size = 4) +
  # Add other labs and theme
  labs(y = "Counts", x = expression(italic(g[2]))) +
  theme_anneke() +

```

```
theme(legend.title=element_blank())
```



We then plotted the bootstrapped g_2 values for both the microsatellite data as well as the SNP array data. For both datasets, the g_2 was significantly different from zero (microsatellite data: $p = 0.041$, SNP array data: $p = 0.001$).

Statistical models - microsatellite heterozygosity

Microsatellite dataset: pup birth mass

We tested for an effect of individual or maternal heterozygosity (sMLH) on pup birth mass with a linear model. Pup and maternal sMLH were included as continuous variables. Covariates were pup sex and year (as factors) and mother age (as continuous variable) (see [Model 1](#)). To make use of a bigger sample size (including pups with known and unknown mothers) we ran the same model while excluding maternal effects (see [Model 2](#)). The results were not significantly different (see [parameter estimates](#)).

Model 1: including maternal effects

```
m1birthmass <- lm(Pup_BirthWeight ~ sMLH_msat39_pup
+ Pup_Sex
+ Year
+ sMLH_msat39_mum
+ Mum_Age,
# (1 | uniqueID_mum), # including or excluding mum ID as random factor does
# not affect the results significantly
```

```

data = pup_data)

summary(m1birthmass)

##
## Call:
## lm(formula = Pup_BirthWeight ~ sMLH_msat39_pup + Pup_Sex + Year +
##     sMLH_msat39_mum + Mum_Age, data = pup_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40697 -0.36247  0.00711  0.28261  1.77869
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.26964    0.40443  10.557 < 2e-16 ***
## sMLH_msat39_pup -0.03216    0.32150   -0.100  0.920379
## Pup_SexM        0.64683    0.05642  11.465 < 2e-16 ***
## Year2018       -0.24587    0.08538   -2.880  0.004241 **
## Year2019        0.09023    0.07433    1.214  0.225593
## Year2020       -0.13157    0.07441   -1.768  0.077921 .
## sMLH_msat39_mum  0.03986    0.31820    0.125  0.900388
## Mum_Age         0.03722    0.01055    3.528  0.000477 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5171 on 333 degrees of freedom
## (673 observations deleted due to missingness)
## Multiple R-squared:  0.3354, Adjusted R-squared:  0.3214
## F-statistic: 24.01 on 7 and 333 DF, p-value: < 2.2e-16

```

Residual check of model

We used the DHARMA package to check three model assumptions: the first figure shows a test for under/overdispersion, the second figure, a QQplot, checks for normality, and the third figure, residuals versus the predictions, allows to check for issues with linearity and equality of error variances. All three tests did not show significant issues.

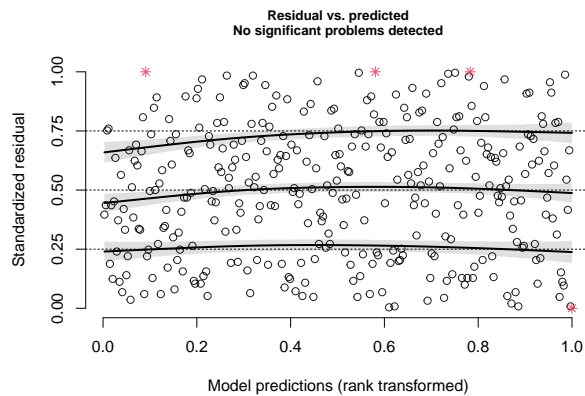
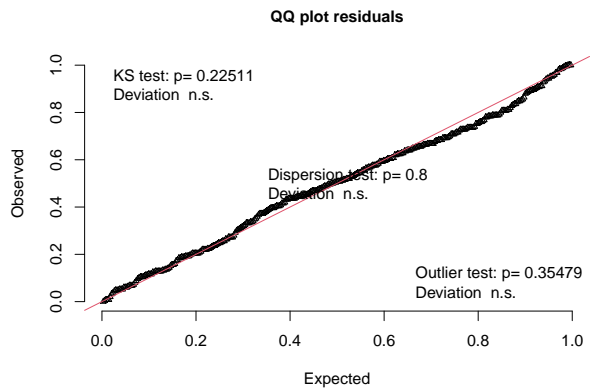
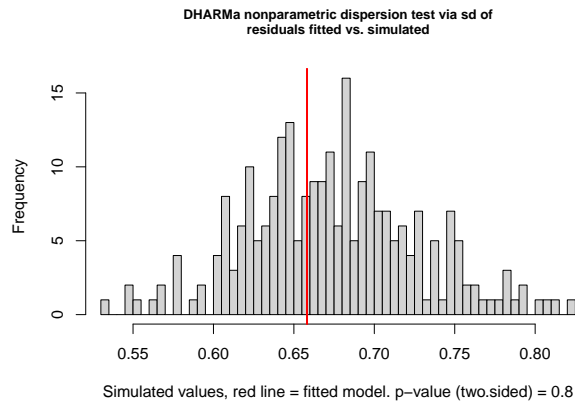
```

# Model assumptions
testDispersion(m1birthmass) # tests for over/underdispersion

##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.97607, p-value = 0.8
## alternative hypothesis: two.sided

plotQQunif(m1birthmass) # detect overall deviations from the expected distribution. Is
  ↪ data normally distributed?
plotResiduals(m1birthmass) # plot residuals against the predicted value

```

Model 2: excluding maternal effects

```
m2birthmass <- lm(Pup_BirthWeight ~ sMLH_msat39_pup
+ Pup_Sex
+ Year,
data = pup_data)

summary(m2birthmass)
```

```
##
## Call:
## lm(formula = Pup_BirthWeight ~ sMLH_msat39_pup + Pup_Sex + Year,
##     data = pup_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53276 -0.35509  0.01509  0.37505  2.32427
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.47752    0.22852  19.594 < 2e-16 ***
## sMLH_msat39_pup 0.17719    0.22502   0.787  0.4312
## Pup_SexM       0.56985    0.04142  13.758 < 2e-16 ***
## Year2018      -0.28974    0.06105  -4.746 2.44e-06 ***
## Year2019       0.13072    0.05577   2.344  0.0193 *
```

```
## Year2020      -0.25487      0.05540  -4.601 4.84e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6078 on 859 degrees of freedom
## (149 observations deleted due to missingness)
## Multiple R-squared:  0.2303, Adjusted R-squared:  0.2258
## F-statistic: 51.4 on 5 and 859 DF, p-value: < 2.2e-16
```

Residual check of model

None of the tests show significant issues with the model.

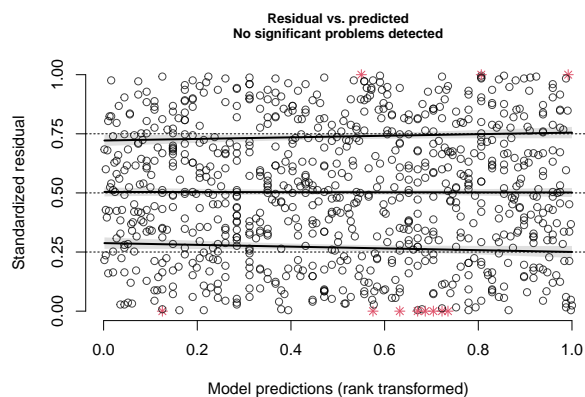
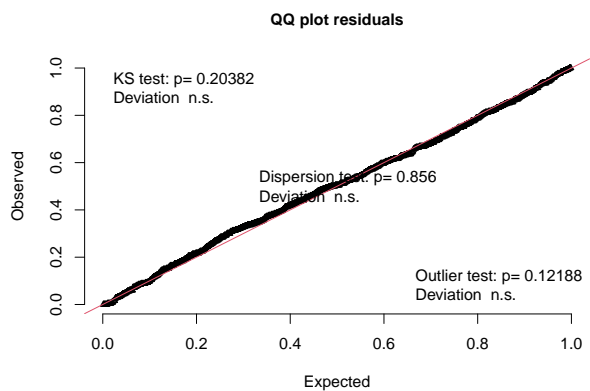
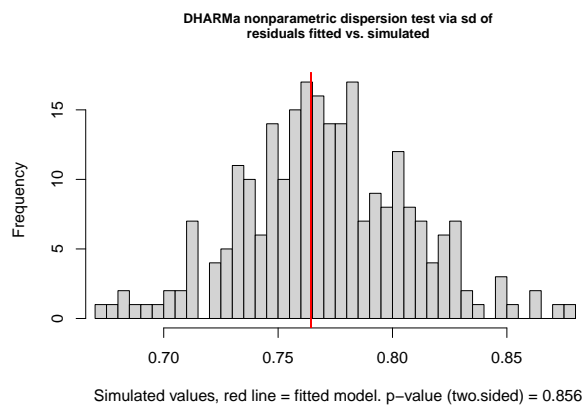
```
# Model assumptions
```

```
testDispersion(m2birthmass)
```

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.9913, p-value = 0.856
## alternative hypothesis: two.sided
```

```
plotQQunif(m2birthmass)
```

```
plotResiduals(m2birthmass)
```



Parameter estimates pup birth mass models

Parameter estimates from: (a) the linear model including maternal genetic diversity (mother sMLH), and (b) the linear model excluding maternal effects. Estimates are shown together with confidence intervals (CI), significant p -values are in bold. For both models, total number of observations, as well as the variance explained by the fixed effects (marginal R^2) and variance explained by both fixed and random effects (conditional R^2) are given. The results of the model including maternal effect (a) compared to the results of the model excluding maternal effects (b) are very similar, even though the sample size is more than double (341 vs. 865, respectively).

```
# Statistical table showing parameter estimates for both models
# Labels
tab_label <- c(
  `(Intercept)` = "Intercept",
  sMLH_msat39_pup = "pup sMLH",
  Pup_SexM = "pup sex [M]",
  sMLH_msat39_mum = "mother sMLH",
  Mum_Age = "mother age",
  Year2018 = "year [2019]",
  Year2019 = "year [2020]",
  Year2020 = "year [2021]")

# Table
# print so it saves but doesn't show the html table,
# which doesn't display nicely in the pdf generated by Rmarkdown
print(tab_model(m1birthmass, m2birthmass,
  pred.labels = tab_label,
  title = "Pup birth mass",
  dv.labels = c("(a) model incl. maternal effect",
    "(b) model excl. maternal effect"),
  show.stat=T,
  string.stat = "t value",
  file = here("Tables", "Table_BW_full_model_vs_no_mat_NEW.html"))

# Make a screenshot of saved html table and save as a png
# so that it can be shown in Rmarkdown pdf
webshot::webshot(here("Tables", "Table_BW_full_model_vs_no_mat_NEW.html"),
  file=here("Tables", "Table_BW_full_model_vs_no_mat_NEW.png"), delay=2,
  ↪ vheight = 450, vwidth = 700)
```

Pup birth mass

| <i>Predictors</i> | (a) model incl. maternal effect | | | | (b) model excl. maternal effect | | | |
|--|---------------------------------|---------------|----------------|----------|---------------------------------|---------------|----------------|----------|
| | <i>Estimates</i> | <i>CI</i> | <i>t value</i> | <i>p</i> | <i>Estimates</i> | <i>CI</i> | <i>t value</i> | <i>p</i> |
| Intercept | 4.27 | 3.47 – 5.07 | 10.56 | <0.001 | 4.48 | 4.03 – 4.93 | 19.59 | <0.001 |
| pup sMLH | -0.03 | -0.66 – 0.60 | -0.10 | 0.920 | 0.18 | -0.26 – 0.62 | 0.79 | 0.431 |
| pup sex [M] | 0.65 | 0.54 – 0.76 | 11.47 | <0.001 | 0.57 | 0.49 – 0.65 | 13.76 | <0.001 |
| year [2019] | -0.25 | -0.41 – -0.08 | -2.88 | 0.004 | -0.29 | -0.41 – -0.17 | -4.75 | <0.001 |
| year [2020] | 0.09 | -0.06 – 0.24 | 1.21 | 0.226 | 0.13 | 0.02 – 0.24 | 2.34 | 0.019 |
| year [2021] | -0.13 | -0.28 – 0.01 | -1.77 | 0.078 | -0.25 | -0.36 – -0.15 | -4.60 | <0.001 |
| mother sMLH | 0.04 | -0.59 – 0.67 | 0.13 | 0.900 | | | | |
| mother age | 0.04 | 0.02 – 0.06 | 3.53 | <0.001 | | | | |
| Observations | 341 | | | | 865 | | | |
| R ² / R ² adjusted | 0.335 / 0.321 | | | | 0.230 / 0.226 | | | |

Microsatellite dataset: survival

Here, we tested for an effect of individual or maternal heterozygosity on pup survival with a generalized linear model (GLM) with a binomial error structure. Survival (1 = survived and 0 = dead) was included as a factor, pup and maternal sMLH were included as continuous variables. Also in this model the addition covariates were pup sex, year (as factors) and mother age (as continuous variable) (see [Model 1](#)). Also here we compared the results with the results generated by running the same model on the bigger dataset (including pups with known and unknown mothers) while excluding maternal effects (see [Model 2](#)). The results were not significantly different (see [parameter estimates](#)), although in the model without maternal effects there seemed to be a year effect that was not present in the model including maternal effects. Pups were more likely to survive in 2019 compared to 2018 but less likely to survive in 2021 compared to 2018.

Model 1: including maternal effects

```

misurvival <- glm(Survival ~ sMLH_msat39_pup
  + Pup_Sex
  + Pup_BirthWeight
  + Year
  + sMLH_msat39_mum
  + Mum_Age,
  #+ (1 | uniqueID_mum), # including or excluding mum ID as random factor does
  # not affect the results significantly
  family=binomial,
  data = pup_data)

```

```
summary(misurvival)
```

```
##
```

```
## Call:
## glm(formula = Survival ~ sMLH_msat39_pup + Pup_Sex + Pup_BirthWeight +
##      Year + sMLH_msat39_mum + Mum_Age, family = binomial, data = pup_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4774   0.3361   0.4556   0.5970   1.1670
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.07018    2.72392   0.393   0.6944
## sMLH_msat39_pup -3.11803    1.83635  -1.698   0.0895 .
## Pup_SexM       -0.91932    0.37143  -2.475   0.0133 *
## Pup_BirthWeight  0.90545    0.31012   2.920   0.0035 **
## Year2018         0.94569    0.54219   1.744   0.0811 .
## Year2019         0.78448    0.46122   1.701   0.0890 .
## Year2020         0.06952    0.38051   0.183   0.8550
## sMLH_msat39_mum -0.92426    1.80433  -0.512   0.6085
## Mum_Age         0.07032    0.06119   1.149   0.2505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 287.77  on 340  degrees of freedom
## Residual deviance: 264.64  on 332  degrees of freedom
## (673 observations deleted due to missingness)
## AIC: 282.64
##
## Number of Fisher Scoring iterations: 5
```

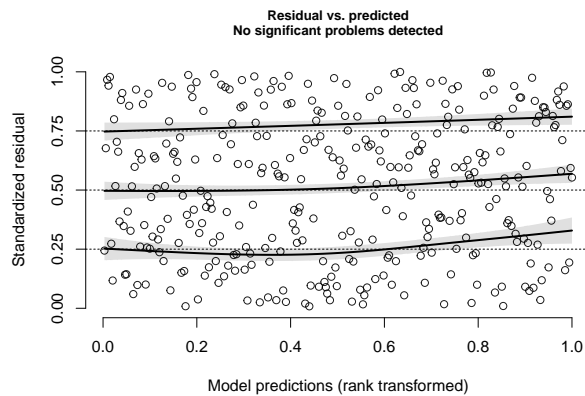
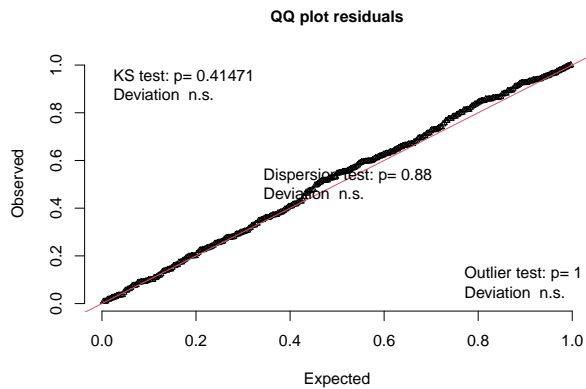
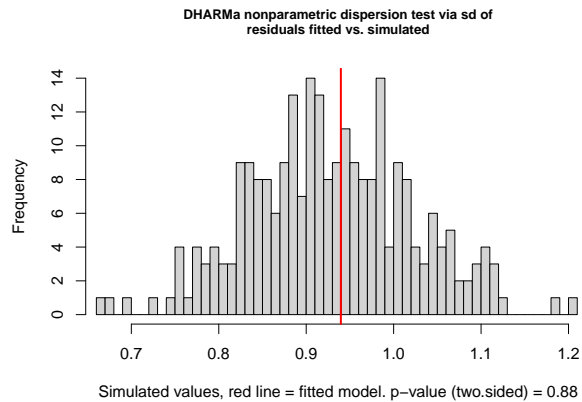
Residual check of model

None of the tests show significant issues with the model.

```
# Model assumptions
testDispersion(m1survival)
```

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data:  simulationOutput
## dispersion = 1.0131, p-value = 0.88
## alternative hypothesis: two.sided
```

```
plotQQunif(m1survival)
plotResiduals(m1survival)
```



Model 2: excluding maternal effects

```
m2survival <- glm(Survival ~ sMLH_msat39_pup
  + Pup_Sex
  + Pup_BirthWeight
  + Year,
  family=binomial,
  data = pup_data)

summary(m2survival)
```

```
##
## Call:
## glm(formula = Survival ~ sMLH_msat39_pup + Pup_Sex + Pup_BirthWeight +
##      Year, family = binomial, data = pup_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6827   0.3092   0.4971   0.6546   1.4968
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.04587    1.25290  -4.027 5.64e-05 ***
## sMLH_msat39_pup  0.93404    0.99080   0.943  0.34583
## Pup_SexM       -0.54304    0.20286  -2.677  0.00743 **
```

```
## Pup_BirthWeight 1.23536 0.16647 7.421 1.16e-13 ***
## Year2018 1.05541 0.32371 3.260 0.00111 **
## Year2019 0.08589 0.26292 0.327 0.74392
## Year2020 -0.51982 0.22989 -2.261 0.02375 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 837.24 on 864 degrees of freedom
## Residual deviance: 739.59 on 858 degrees of freedom
## (149 observations deleted due to missingness)
## AIC: 753.59
##
## Number of Fisher Scoring iterations: 5
```

Residual check of model

None of the tests show significant issues with the model.

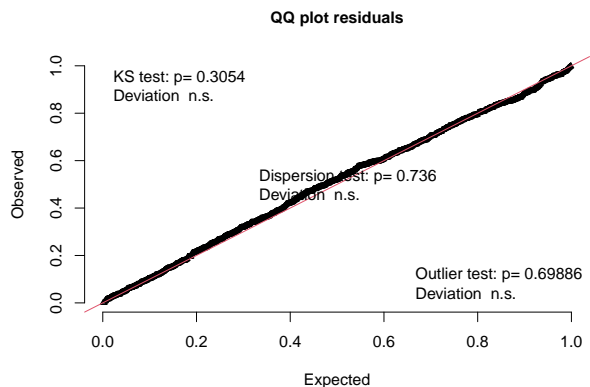
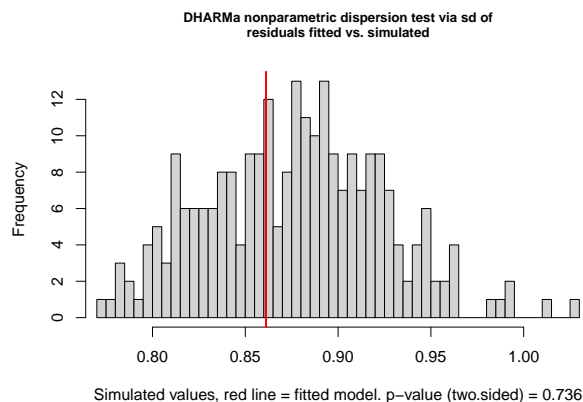
```
# Model assumptions
```

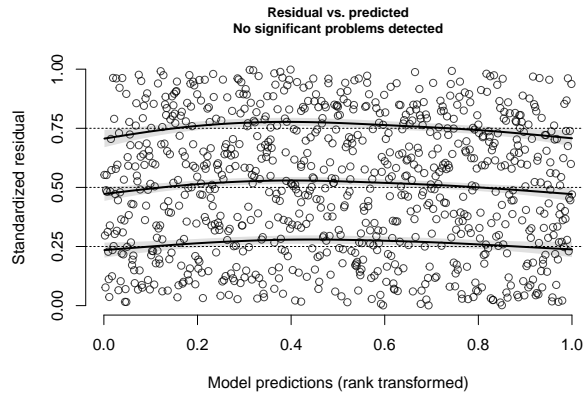
```
testDispersion(m2survival)
```

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.98214, p-value = 0.736
## alternative hypothesis: two.sided
```

```
plotQQunif(m2survival)
```

```
plotResiduals(m2survival)
```





Parameter estimates pup survival models

Parameter estimates from: (a) the GLM with a binomial error distribution including maternal genetic diversity, and (b) the linear model excluding maternal effects. Estimates are shown together with confidence intervals (CI), significant p -values are in bold. For both models, total number of observations, as well as the variance explained by the fixed effects (marginal R^2) and variance explained by both fixed and random effects (conditional R^2) are given. The results of the model including maternal effect (a) compared to the results of the model excluding maternal effects (b) are very similar, even though the sample size is more than double (341 vs. 865, respectively).

```
# Labels
tab_labSurvival <- c(
  `(Intercept)` = "Intercept",
  sMLH_msat39_pup = "pup sMLH",
  Pup_BirthWeight = "pup birth mass",
  Pup_SexM = "pup sex [M]",
  sMLH_msat39_mum = "mother sMLH",
  Mum_Age = "mother age",
  Year2018 = "year [2019]",
  Year2019 = "year [2020]",
  Year2020 = "year [2021]")

# Table
print(tab_model(m1survival, m2survival,
  pred.labels = tab_labSurvival,
  title = "Pup survival",
  dv.labels = c("(a) model incl. maternal effect",
    "(b) model excl. maternal effect"),
  #order.terms = c(1, 2, 4, 3, 7, 5, 6),
  show.stat=T,
  string.stat = "t value",
  file = here("Tables", "Table_survival_full_model_vs_no_mat.html"))

# Makes a screenshot of saved html table and saves as a png
webshot::webshot(here("Tables", "Table_survival_full_model_vs_no_mat.html"),
  file=here("Tables", "Table_survival_full_model_vs_no_mat.png"), delay=2,
  vheight = 450, vwidth = 700)
```


Pup survival

| <i>Predictors</i> | (a) model incl. maternal effect | | | | (b) model excl. maternal effect | | | |
|---------------------|---------------------------------|---------------|----------------|--------------|---------------------------------|--------------|----------------|--------------|
| | <i>Odds Ratios</i> | <i>CI</i> | <i>t value</i> | <i>p</i> | <i>Odds Ratios</i> | <i>CI</i> | <i>t value</i> | <i>p</i> |
| Intercept | 2.92 | 0.01 – 627.88 | 0.39 | 0.694 | 0.01 | 0.00 – 0.07 | -4.03 | <0.001 |
| pup sMLH | 0.04 | 0.00 – 1.56 | -1.70 | 0.090 | 2.54 | 0.36 – 17.80 | 0.94 | 0.346 |
| pup sex [M] | 0.40 | 0.19 – 0.82 | -2.48 | 0.013 | 0.58 | 0.39 – 0.86 | -2.68 | 0.007 |
| pup birth mass | 2.47 | 1.36 – 4.63 | 2.92 | 0.004 | 3.44 | 2.50 – 4.81 | 7.42 | <0.001 |
| year [2019] | 2.57 | 0.95 – 8.28 | 1.74 | 0.081 | 2.87 | 1.55 – 5.56 | 3.26 | 0.001 |
| year [2020] | 2.19 | 0.92 – 5.71 | 1.70 | 0.089 | 1.09 | 0.65 – 1.84 | 0.33 | 0.744 |
| year [2021] | 1.07 | 0.51 – 2.29 | 0.18 | 0.855 | 0.59 | 0.38 – 0.93 | -2.26 | 0.024 |
| mother sMLH | 0.40 | 0.01 – 13.36 | -0.51 | 0.608 | | | | |
| mother age | 1.07 | 0.95 – 1.22 | 1.15 | 0.250 | | | | |
| Observations | 341 | | | | 865 | | | |
| R ² Tjur | 0.069 | | | | 0.131 | | | |

Microsatellite dataset: pup weight gain

In order to test for weight gain, the data needs to be filtered to include only surviving pups (for most pups that did not survive, a second weight was not taken, and therefore weight gain could not be calculated).

We then tested for an effect of individual or maternal heterozygosity on pup survival with a linear mixed model (LMM) using the restricted maximum-likelihood (REML) approach. Growth (difference in weight between birth and recapture) was included as a continuous variable. Addition covariates were pup sex, year and mother age (see [Model 1](#)). Pup age (number of days between birth and recapture) was included in as a random effect to control for the variation in pup age.

Again, here we compared the results with the results generated by running the same model on the bigger dataset (including pups with known and unknown mothers) while excluding maternal effects (see [Model 2](#)). The results were comparable between the two models (see [parameter estimates](#)), although in the model without maternal effects there seemed to be an effect of both pup sex and birth mass, where males gained more weight than females and pups that were born heavier also gained more weight. In the model with maternal effect, these effects were not present, however, we find here a small significant effect of pup sMLH, which is not present in the model without maternal effects.

```
# Keep only pups that survived
pup_alive <- pup_data %>%
  filter(Survival == "1") %>%
  select(-c(Pup_Death, Survival))
```

Model 1: including maternal effects

```
migrowth <- lmerTest::lmer(WeightGain ~ sMLH_msat39_pup
  + Pup_Sex
```

```

+ Pup_BirthWeight
+ Year
+ sMLH_msat39_mum
+ Mum_Age
+ (1 | Age_Tag),
#+ (1 | uniqueID_mum), # including or excluding mum ID as
↳ random factor does not affect the results significantly
data = pup_alife)

```

```
summary(m1growth)
```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: WeightGain ~ sMLH_msat39_pup + Pup_Sex + Pup_BirthWeight + Year +
##       sMLH_msat39_mum + Mum_Age + (1 | Age_Tag)
## Data: pup_alife
##
## REML criterion at convergence: 992
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.2502 -0.6279  0.0561  0.5781  2.5040
##
## Random effects:
##  Groups      Name                Variance Std.Dev.
##  Age_Tag      (Intercept) 0.2831    0.532
##  Residual                1.5913    1.261
## Number of obs: 290, groups: Age_Tag, 46
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    0.56877    1.28662 271.27527   0.442   0.6588
## sMLH_msat39_pup 2.21274    0.87405 260.73777   2.532   0.0119 *
## Pup_SexM        0.35312    0.19481 277.22811   1.813   0.0710 .
## Pup_BirthWeight 0.04295    0.15982 275.30120   0.269   0.7883
## Year2018        1.47865    0.23705 272.47639   6.238 1.69e-09 ***
## Year2019        2.19757    0.20812 278.65049  10.559 < 2e-16 ***
## Year2020        1.66577    0.21825 279.36539   7.632 3.66e-13 ***
## sMLH_msat39_mum -1.06241    0.86121 264.87430  -1.234   0.2184
## Mum_Age         0.07830    0.03310 280.77285   2.366   0.0187 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual check of model

None of the tests show significant issues with the model.

```

# Model assumptions
testDispersion(m1growth)

```

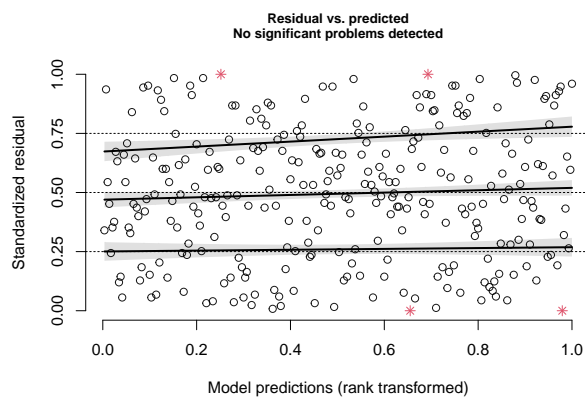
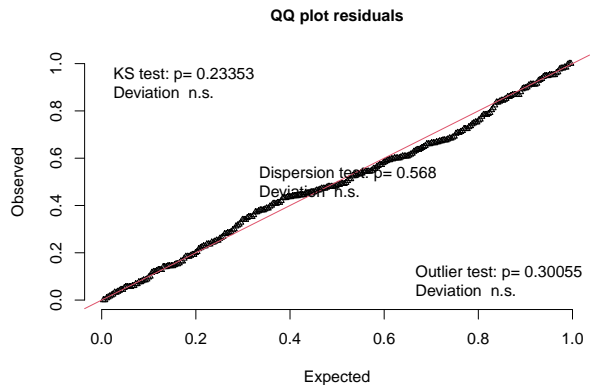
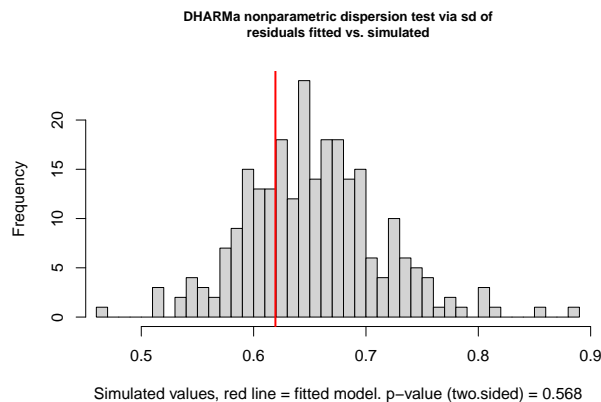
```

##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated

```

```
##
## data: simulationOutput
## dispersion = 0.94834, p-value = 0.568
## alternative hypothesis: two.sided
```

```
plotQQunif(m1growth)
plotResiduals(m1growth)
```



Model 2: excluding maternal effects

```
m2growth <- lmerTest::lmer(WeightGain ~ sMLH_msat39_pup
+ Pup_Sex
+ Pup_BirthWeight
+ Year
+ (1 | Age_Tag),
data = pup_alive)

summary(m2growth)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: WeightGain ~ sMLH_msat39_pup + Pup_Sex + Pup_BirthWeight + Year +
## (1 | Age_Tag)
## Data: pup_alive
##
## REML criterion at convergence: 2535.1
```

```
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.2421 -0.6164  0.0191  0.5914  3.2394
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   Age_Tag  (Intercept) 0.4162   0.6452
##   Residual              1.9610   1.4004
## Number of obs: 702, groups: Age_Tag, 64
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    0.79612    0.75700  681.50859   1.052  0.29332
## sMLH_msat39_pup 0.44197    0.59754  654.61408   0.740  0.45978
## Pup_SexM       0.37776    0.12340  651.20921   3.061  0.00230 **
## Pup_BirthWeight 0.28487    0.09754  660.31330   2.921  0.00361 **
## Year2018       1.71760    0.15841  659.91879  10.843 < 2e-16 ***
## Year2019       1.94567    0.14878  690.02196  13.077 < 2e-16 ***
## Year2020       1.59356    0.15736  688.17459  10.127 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

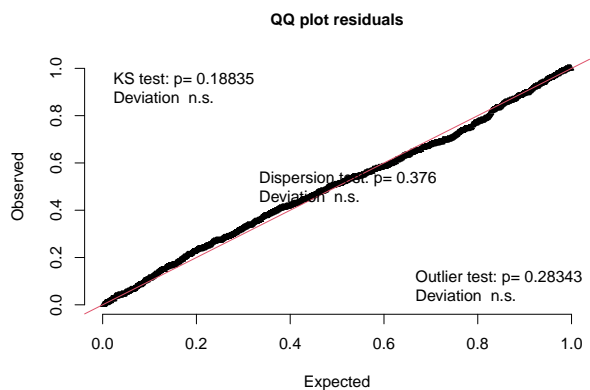
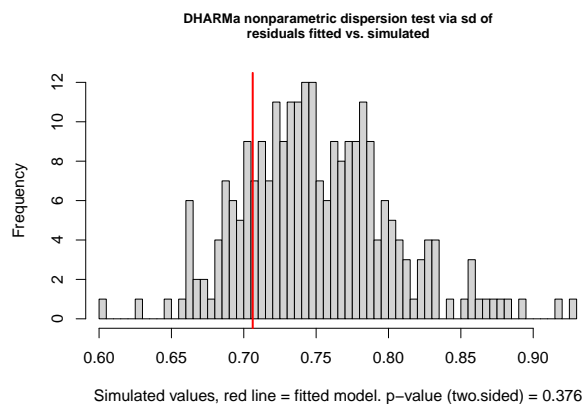
Residual check of model

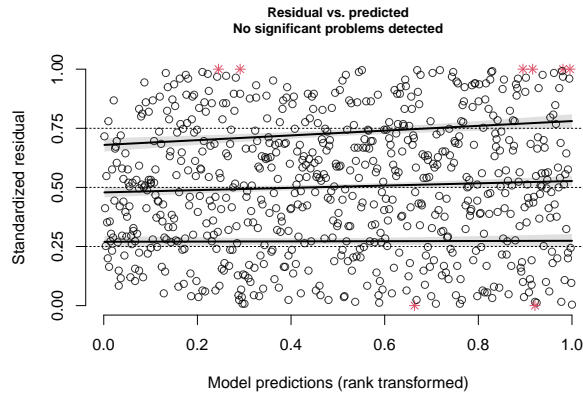
None of the tests show significant issues with the model.

```
# Model assumptions
testDispersion(m2growth)
```

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.94053, p-value = 0.376
## alternative hypothesis: two.sided
```

```
plotQQunif(m2growth)
plotResiduals(m2growth)
```





Parameter estimates pup survival models

Parameter estimates from: (a) the REML linear mixed model including maternal genetic diversity, and (b) excluding maternal effects. Estimates are shown together with confidence intervals (CI), significant p -values are in bold. For both models, total number of observations, as well as the variance explained by the fixed effects (marginal R^2) and variance explained by both fixed and random effects (conditional R^2) are given. Mother ID was included as a random effect to account for repeated measures in model A. For this random effect, the mean squared error (σ^2), between group variance (τ_{00}), Intraclass Correlation Coefficient (ICC; the consistency within an individual across multiple measurements) and the sample size (n) are reported.

```
# Labels
tab_labGrowth <- c(
  `(Intercept)` = "Intercept",
  sMLH_msat39_pup = "pup sMLH",
  Pup_BirthWeight = "pup birth mass",
  Pup_SexM = "pup sex [M]",
  sMLH_msat39_mum = "mother sMLH",
  Mum_Age = "mother age",
  Year2018 = "year [2019]",
  Year2019 = "year [2020]",
  Year2020 = "year [2021]")

# Table
print(tab_model(m1growth, m2growth,
  pred.labels = tab_labGrowth,
  title = "Pup growth",
  dv.labels = c("(a) model incl. maternal effect",
    "(b) model excl. maternal effect"),
  #order.terms = c(1, 2, 4, 3, 7, 5, 6),
  show.stat=T,
  string.stat = "t value",
  file = here("Tables", "Table_growth_full_model_vs_no_mat_NEW.html"))

# Makes a screenshot of saved html table and saves as a png
webshot::webshot(here("Tables", "Table_growth_full_model_vs_no_mat_NEW.html"),
  file=here("Tables", "Table_growth_full_model_vs_no_mat_NEW.png"),
  ↵ delay=2, vheight = 450, vwidth = 750)
```

Pup growth

| <i>Predictors</i> | (a) model incl. maternal effect | | | | (b) model excl. maternal effect | | | |
|--|--|--------------|----------------|------------------|--|--------------|----------------|------------------|
| | <i>Estimates</i> | <i>CI</i> | <i>t value</i> | <i>p</i> | <i>Estimates</i> | <i>CI</i> | <i>t value</i> | <i>p</i> |
| Intercept | 0.57 | -1.96 – 3.10 | 0.44 | 0.659 | 0.80 | -0.69 – 2.28 | 1.05 | 0.293 |
| pup sMLH | 2.21 | 0.49 – 3.93 | 2.53 | 0.012 | 0.44 | -0.73 – 1.62 | 0.74 | 0.460 |
| pup sex [M] | 0.35 | -0.03 – 0.74 | 1.81 | 0.071 | 0.38 | 0.14 – 0.62 | 3.06 | 0.002 |
| pup birth mass | 0.04 | -0.27 – 0.36 | 0.27 | 0.788 | 0.28 | 0.09 – 0.48 | 2.92 | 0.004 |
| year [2019] | 1.48 | 1.01 – 1.95 | 6.24 | <0.001 | 1.72 | 1.41 – 2.03 | 10.84 | <0.001 |
| year [2020] | 2.20 | 1.79 – 2.61 | 10.56 | <0.001 | 1.95 | 1.65 – 2.24 | 13.08 | <0.001 |
| year [2021] | 1.67 | 1.24 – 2.10 | 7.63 | <0.001 | 1.59 | 1.28 – 1.90 | 10.13 | <0.001 |
| mother sMLH | -1.06 | -2.76 – 0.63 | -1.23 | 0.218 | | | | |
| mother age | 0.08 | 0.01 – 0.14 | 2.37 | 0.019 | | | | |
| Random Effects | | | | | | | | |
| σ^2 | 1.59 | | | | 1.96 | | | |
| τ_{00} | 0.28 | Age_Tag | | | 0.42 | Age_Tag | | |
| ICC | 0.15 | | | | 0.18 | | | |
| N | 46 | Age_Tag | | | 64 | Age_Tag | | |
| Observations | 290 | | | | 702 | | | |
| Marginal R ² / Conditional R ² | 0.342 / 0.441 | | | | 0.248 / 0.380 | | | |

Statistical models - SNP inbreeding

Birth weight analysis

In this section of the script factors affecting birth weight are investigated.

Filtered data for birth weight analysis For both birth weight analyses and later growth analyses only pups that survived through the season are included. Mums that were no genetic match with their pups were also removed (concerns pup H2 and H5, they were switched and suckled by the others mum).

Data visualization Before fitting the models, the raw data is visualized to explore distribution.

```
#min/mean/max of birth weight
mean_birth_weight <- mean(UniqueSurvivors_Day60$Birth_weight, na.rm = T)
#5.721333
#min_birth_weight <- min(UniqueSurvivors_Day60$Birth_weight, na.rm = T)
#4.2
#max_birth_weight <- max(UniqueSurvivors_Day60$Birth_weight, na.rm = T)
#7.8

birth_weight_raw <- ggplot(UniqueSurvivors_Day60, aes(x = Birth_weight)) +
  geom_histogram(binwidth = 0.2, fill = "orange", color = "black") +
  geom_vline(aes(xintercept = mean_birth_weight), color = "purple", linetype = "dashed",
    ↪ linewidth = 0.8) +
  labs(title = "Birth weight distribution",
    ↪ x = "Weight (kg)",
    ↪ y = "Frequency") +
  theme_minimal()
#Based on visual inspection, the birth weight looks fairly normally distributed

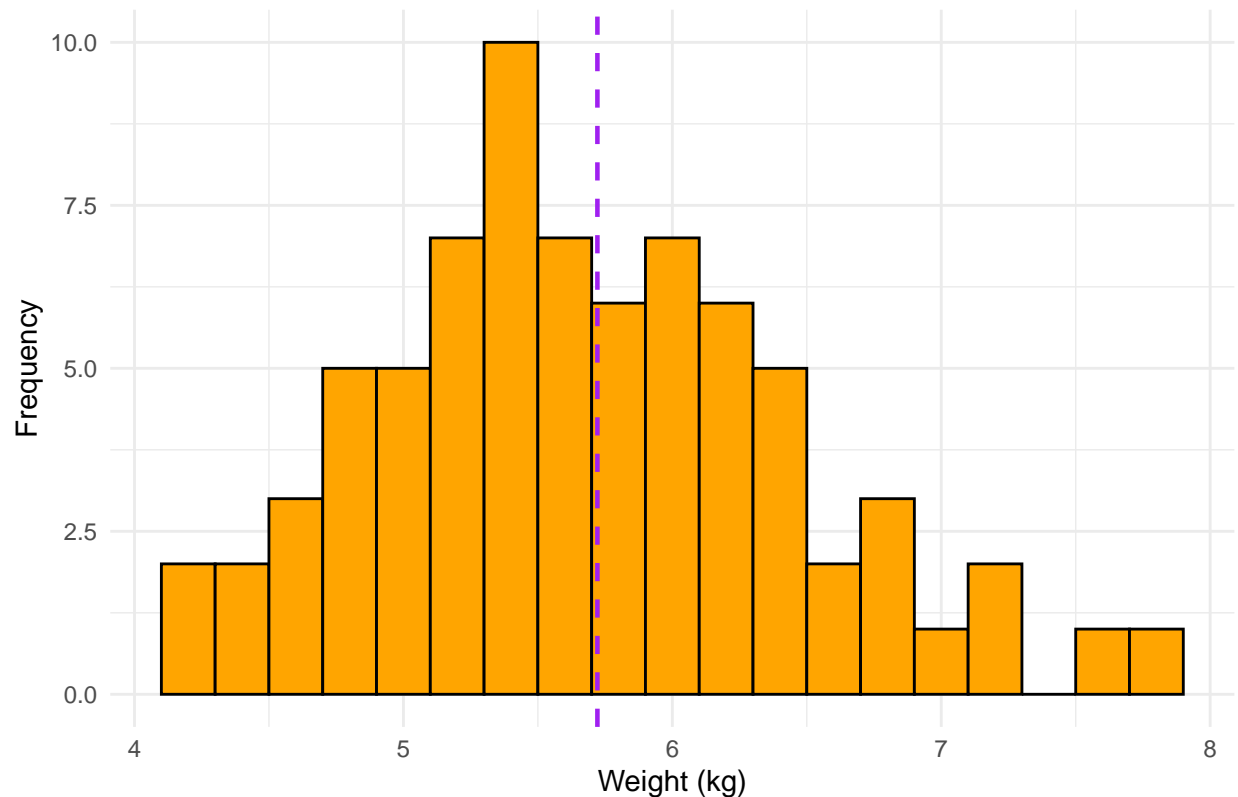
shapiro.test(UniqueSurvivors_Day60$Birth_weight) #the shapiro-wilk test tests for
↪ normality. The null-hypothesis is that the population is normally distributed. P
↪ value > 0.05 implying that the distribution of the data are not significantly
↪ different from normal distribution. Therefore, we can assume normality.

##
## Shapiro-Wilk normality test
##
## data: UniqueSurvivors_Day60$Birth_weight
## W = 0.98272, p-value = 0.3983

#plots
plot_grid(birth_weight_raw, labels = "AUTO", label_size = 15)

## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```

A Birth weight distribution



Model The birth weight data is normally distributed, so the model is built with a gaussian distribution.

```
BW.model <- lm(Birth_weight ~ Sex +
               Season +
               Beach +
               sMLH_SNP.new +
               sMLH_SNP.new_mum,
               data = UniqueSurvivors_Day60)

#plot(BW.model)
#summary(BW.model)
#testDispersion(BW.model) #good fit
#plotQQunif(BW.model) #deviation not significant

#Test model fit
#Predict values for your existing data
predicted_values <- predict(BW.model, type = "response")
na_values <- rep(NA, 4) #to make the predicted no. of values equal to the observed data
predicted_values <- c(predicted_values, na_values)

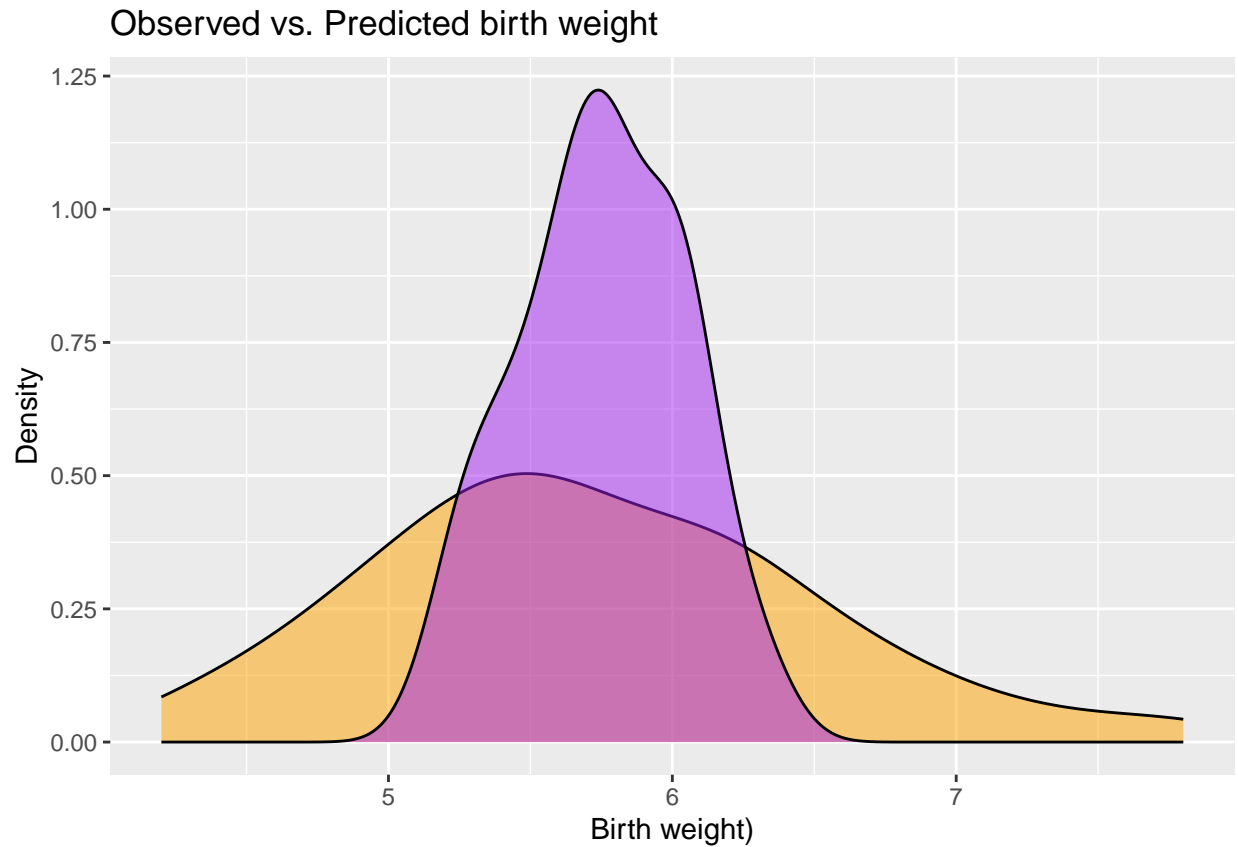
#plot the densities of the predicted and observed data
BW_model <- ggplot(UniqueSurvivors_Day60, aes(x = Birth_weight)) +
  geom_density(fill = "orange", alpha = 0.5) +
  geom_density(aes(x = predicted_values), fill = "purple", alpha = 0.5) +
  labs(x = "Birth weight", y = "Density") +
  ggtitle("Observed vs. Predicted birth weight")
```



```
BW_model #Note: densities does not match greatly
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_density()`).
```



```
sjPlot::tab_model(BW.model)
```

Birth weight

Predictors

Estimates

CI

p

(Intercept)

17.23

-1.35 – 35.82

0.069

Sex [M]

0.37

-0.06 – 0.80
 0.094
 Season [1920]
 0.30
 -0.05 – 0.66
 0.090
 Beach [SSB]
 0.15
 -0.20 – 0.51
 0.390
 sMLH SNP new
 -1.51
 -15.98 – 12.96
 0.836
 sMLH SNP new mum
 -10.38
 -24.24 – 3.48
 0.140
 Observations
 72
 R2 / R2 adjusted
 0.143 / 0.078

```

#model with froh
BW.model.froh <- lm(Birth_weight ~ froh +
                    froh_mum +
                    Sex +
                    Season +
                    Beach,
                    data = UniqueSurvivors_Day60)

#plot(BW.model.froh)
#summary(BW.model.froh)
#testDispersion(BW.model.froh) #good fit
#plotQQunif(BW.model.froh) #deviation not significant

#sjPlot::tab_model(BW.model.froh)
  
```

Survival analysis

Note: for the pups C20 and N1 their mums, F20 and FWB1, respectively, died during the sampling season as well.

First step for the survival analysis is to transform the 'Death' variable into a binary (0,1) variable and remove the two pups that were not cared for by their biological mum.

```
Unique_Day60$Survived <- ifelse(Unique_Day60$Death == 'N',1,0)
```

Binomial data for survival analysis

Model The model is built with a binary response variable. The model includes includes sMLH values calculated from SNP data.

```
Survival.model <- glm(Survived ~ sMLH_SNP.new +  
  Sex +  
  Birth_weight +  
  Season +  
  Beach +  
  sMLH_SNP.new_mum, data = Unique_Day60, family = 'binomial')  
  
#summary(Survival.model)  
#Binary variable codes so 0 means the pup died that season and 1 means the pup survived.  
#Birth weight and beach is significant  
  
#Test model  
#testDispersion(Survival.model) #good fit  
#plotQQunif(Survival.model) #deviation not significant  
  
#S.model <- simulateResiduals(fittedModel = Survival.model, plot = F)  
#plot(S.model)  
  
sjPlot::tab_model(Survival.model)
```

Survived

Predictors

Odds Ratios

CI

P

(Intercept)

0.00

0.00 – 370698589295009267722.00

0.536

sMLH SNP new

0.00

0.00 – 9279502535017.64

0.487

Sex [M]

0.92

0.24 – 3.37

0.904

Birth weight

2.28

1.11 – 5.54

0.041

Season [1920]

0.89

0.27 – 2.95

0.854

Beach [SSB]

4.57

1.39 – 18.04

0.018

sMLH SNP new mum

780628827631272.12

0.00 – 10530105895299114617862688224684886288.00

0.169

Observations

91

R2 Tjur

0.156

```
#survival model with froh
Survival.model.froh <- glm(Survived ~ froh +
  froh_mum +
  Sex +
  Season +
  Beach +
  Birth_weight,
  data = Unique_Day60, family = 'binomial')

#summary(Survival.model.froh)
#Binary variable codes so 0 means the pup died that season and 1 means the pup survived.
#Birth weight and beach is significant

#Test model
#testDispersion(Survival.model.froh) #good fit
#plotQQunif(Survival.model.froh) #deviation not significant

sjPlot::tab_model(Survival.model.froh)
```

Survived

| |
|---|
| Predictors |
| Odds Ratios |
| CI |
| P |
| (Intercept) |
| 0.01 |
| 0.00 – 9.37 |
| 0.216 |
| froh |
| 3759442126998.22 |
| 0.00 – 2561652649118132075204480622686600220.00 |
| 0.277 |
| froh mum |
| 0.00 |
| 0.00 – 128696014461957.86 |
| 0.469 |
| Sex [M] |
| 1.12 |
| 0.30 – 4.05 |
| 0.867 |
| Season [1920] |
| 0.88 |
| 0.27 – 2.76 |
| 0.821 |
| Beach [SSB] |
| 3.91 |
| 1.23 – 14.56 |
| 0.028 |
| Birth weight |
| 2.22 |
| 1.10 – 5.22 |
| 0.042 |
| Observations |
| 91 |
| R2 Tjur |
| 0.153 |

Growth curves with repeated measures

The dataset from FWB and SSB in season 1819/1920 contains repeated measures of growth at 6 different time points until tagging. This section contains the analysis of growth based on growth curves utilizing the repeated measures. *RM = repeated measures*

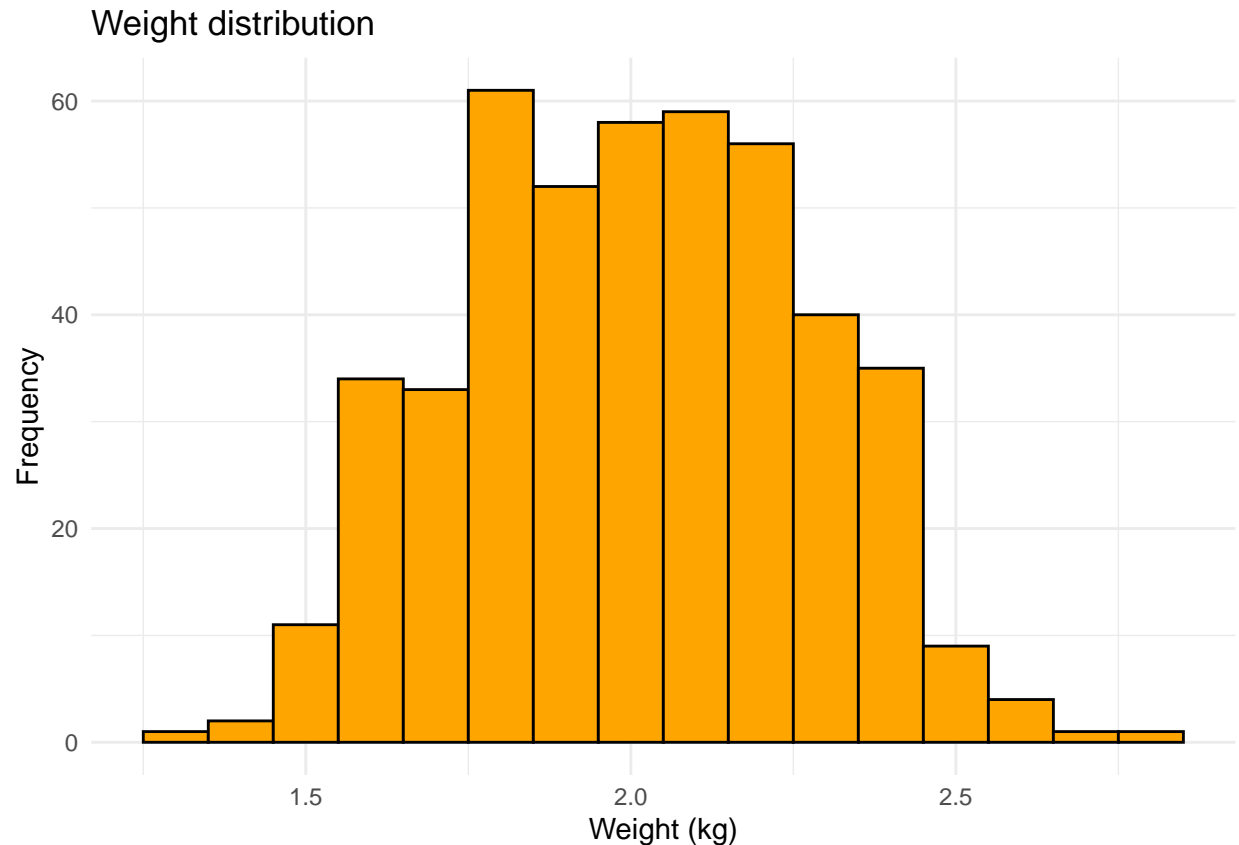
Data visualization Before fitting the models, the raw data is visualized to explore distribution. Weight data is visually expected on its own and fitted against age in days to understand general distribution and pattern over time. The raw data is visually expected using ggplot2 and further explored using the shapiro-wilk test.

```
#mean of weight_kg
mean_weight <- mean(SurvivorsRM_Day60$Weight_kg, na.rm = T)
#7.684354
#max_tagging_weight <- max(SurvivorsRM_Day60$Last_weight, na.rm = T)
#16.3

weight_raw <- ggplot(SurvivorsRM_Day60, aes(x = Weight_kg)) +
  geom_histogram(binwidth = 0.4, fill = "orange", color = "black") +
  geom_vline(aes(xintercept = mean_weight), color = "purple", linetype = "dashed",
    ↪ linewidth = 0.8) +
  labs(title = "Weight distribution",
    x = "Weight (kg)",
    y = "Frequency") +
  theme_minimal()
#Based on visual inspection, data is slightly right-skewed

ggplot(SurvivorsRM_Day60, aes(x = log(Weight_kg))) +
  geom_histogram(binwidth = 0.1, fill = "orange", color = "black") +
  labs(title = "Weight distribution",
    x = "Weight (kg)",
    y = "Frequency") +
  theme_minimal()
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```



```
shapiro.test(SurvivorsRM_Day60$Weight_kg) #the shapiro-wilk test tests for normality. The
→ null-hypothesis is that the population is normally distributed. The test is
→ significant, so we reject the null-hypothesis. The weight_kg data is not normally
→ distributed. By scaling the outcome variable, we can make it more suitable to built
→ models with, even though it does not necessarily normalize the distribution.
```

```
##
## Shapiro-Wilk normality test
##
## data: SurvivorsRM_Day60$Weight_kg
## W = 0.96611, p-value = 8.89e-09
```

```
#shapiro.test(scale(SurvivorsRM_Day60$Weight_kg)) still not normally distributed, however
→ if it allows for normally distributed residuals down stream, it is valid.
#SurvivorsRM_Day60$Weight_kg_scale <- scale(SurvivorsRM_Day60$Weight_kg)
SurvivorsRM_Day60$Age_Days_scale <- scale(SurvivorsRM_Day60$Age_Days)
#shapiro.test(log(SurvivorsRM_Day60$Weight_kg)) #a log transformation could handle this,
→ if needed
#however, GLM with appropriate data distribution or nonlinear models like logistic and
→ gompertz are also a good way to solve this.
SurvivorsRM_Day60$Weight_kg_log <- log(SurvivorsRM_Day60$Weight_kg)

weight_over_time <- ggplot(SurvivorsRM_Day60, aes(x = Age_Days, y = Weight_kg)) +
  geom_point() +
  labs(title = "Weight gain over time",
       x = "Days",
```

```

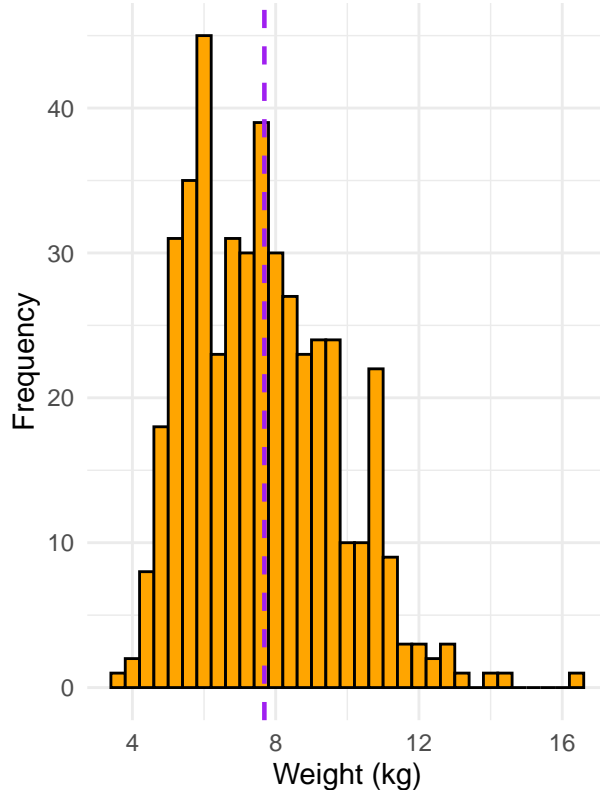
y = "Weight (kg)" +
  theme_minimal()

#plots
plot_grid(weight_raw, weight_over_time, labels = "AUTO", label_size = 20)

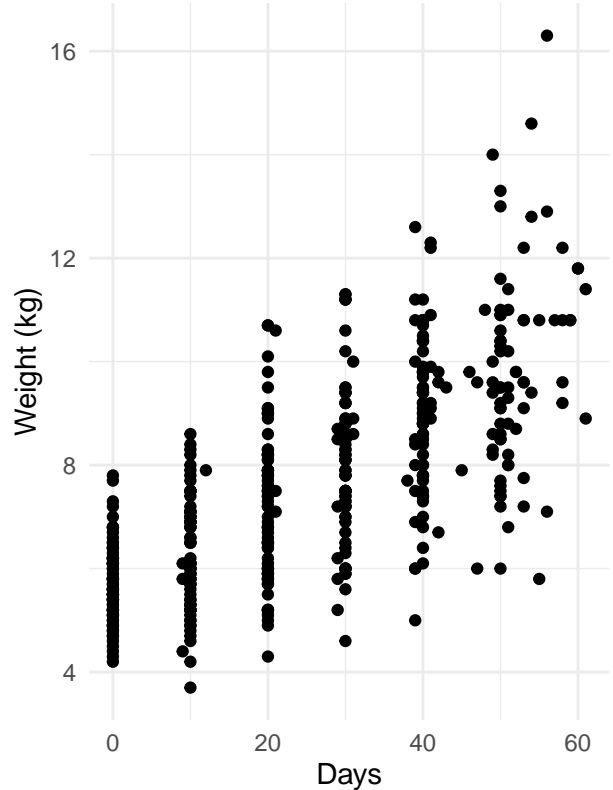
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 2 rows containing missing values (`geom_point()`).

```

A Weight distribution



B Weight gain over time



Based on both visual inspection and the shapiro-wilk test, the weight data does not follow a normal distribution but is slightly right-skewed. The weight data follows a gamma distribution and will be modelled as such.

Repeated measures growth curve models

For the models with repeated measures, a random effects term 'ID' is added to inform the model that some observations are clustered within the same individual. The base is built around a generalized linear mixed-effects model (GLMM) to account for the repeated measures and the right-skewed data. The Gamma(link = "log") data distribution ensures that the predicted values are positive.

```

#unconditional means model aka null model (no predictors)
RMO <- lmer(Weight_kg_log ~ 1 + (1 | ID),
            data = SurvivorsRM_Day60,
            REML = F)

#summary(RMO)

#growth varying per day, each individual has different intercepts, but the same slope

```



```

RM <- lmer(Weight_kg_log ~ Age_Days_scale + (1 | ID),
          data = SurvivorsRM_Day60,
          REML = F)
#summary(RM)

#growth varying per day, each individual has different slopes ((Age_Days | ID))
RM1 <- lmer(Weight_kg_log ~ Age_Days_scale + (Age_Days_scale - 1 | ID),
          data = SurvivorsRM_Day60,
          REML = F)

## boundary (singular) fit: see ?isSingular

#summary(RM1)

#growth varying per day, both intercept and slope vary per individual
RM2 <- lmer(Weight_kg_log ~ Age_Days_scale + (1 + Age_Days_scale | ID),
          data = SurvivorsRM_Day60,
          REML = F)
#summary(RM2)

#explanation: Fixed effects estimate: Intercept is average birth weight
# Age_Days is one day change in weight
# Random effects: associated variance
# Intercept: how much variance in birth weight between pups
# Age_Days: difference in slope. 0.00060 might not sound as much
# Corr: the correlation between the slope and intercept: 0.36
# it's positive, which indicates that pups with higher intercept
# on average has a steeper slope as well.
# Residual: 0.5593029. Refers to the variance not explained by the variables
↪ in the model
# additionally, looking at confint(RM2), the confidence interval
# for the intercept and for the Age_Days does not cross 0.
#ICC = repeatability of weight across individuals

models <- list(RM0, RM, RM1, RM2)
mod.names <- c('null-model', 'intercept', 'slope', 'i+s')
AIC_model_structure <- aictab(cand.set = models, modnames = mod.names)
AIC_model_structure

##
## Model selection based on AICc:
##
##           K      AICc Delta_AICc AICcWt Cum.Wt      LL
## i+s        6 -561.00      0.00      1      1 286.59
## intercept  4 -542.06     18.93      0      1 275.08
## slope      4 -259.78    301.22      0      1 133.94
## null-model 3   32.95    593.94      0      1 -13.45

#the RM2 model, which allows both intercept and slope to vary per individual performs
↪ best.
#testDispersion(RM2)
#plotQQunif(RM2)

#clean up

```

```
rm(RM0, RM1, RM, models, mod.names)
```

Exploration of growth curve fit We can explore other growth curves. In the larger dataset birth weight ranges from 2.45 to 7.7 kg and weight at day 60 from 6.5 to 14.6 kg. In this dataset with repeated measures, lowest birth weight is 4.2 and highest last weight is 16.3.

```
#For these models, the parameters K, r and t are used to fit the model
#K is the max weight the pups can reach, set to 20kg, as heaviest pup was
↪ 16.3
#r is the growth rate, the average growth rate is used: 0.080821
#t is the inflection point; the time at which the pups growth most rapidly.
#set at 30days, as this is the mid-point

#logistic growth model
logistic.model <- nls(Weight_kg ~ K / (1 + exp(-r * (Age_Days - t))),
  data = SurvivorsRM_Day60,
  start = list(K = 20, r = 0.080821, t = 30))

#gompertz growth model
gompertz.model <- nls(Weight_kg ~ K * exp(-exp(-r * (Age_Days - t))),
  data = SurvivorsRM_Day60,
  start = list(K = 20, r = 0.080821, t = 30))

#Linear
linear.model <- nls(Weight_kg ~ a * Age_Days + b,
  data = SurvivorsRM_Day60,
  start = list(a = 0.080821, b = 2.45))

#for the linear model, a and b are parameters used to fit the model
#a is the linear growth rate; the average growth rate is used:
↪ 0.080821
#b is the birth weight, here set at 2.45kg, as lightest measured pup
↪ was 2.45kg.

# Compare the models using AIC
models.1 <- list(linear.model, logistic.model, gompertz.model)
mod.names.1 <- c('linear.model', 'logistic.model', 'gompertz.model')
AIC_growth_curves <- aictab(cand.set = models.1, modnames = mod.names.1)
AIC_growth_curves

##
## Model selection based on AICc:
##
##           K    AICc Delta_AICc AICcWt Cum.Wt      LL
## linear.model 3 1623.24      0.00  0.52  0.52 -808.59
## logistic.model 4 1624.81      1.57  0.24  0.76 -808.36
## gompertz.model 4 1624.82      1.58  0.24  1.00 -808.36

#in this basic model format, the linear model is the best fit

# Visualize the fits
ggplot(SurvivorsRM_Day60, aes(x = Age_Days, y = Weight_kg)) +
  geom_point() +
  geom_smooth(method = "nls", formula = y ~ a * x + b, se = FALSE, color = "orange",
```

```

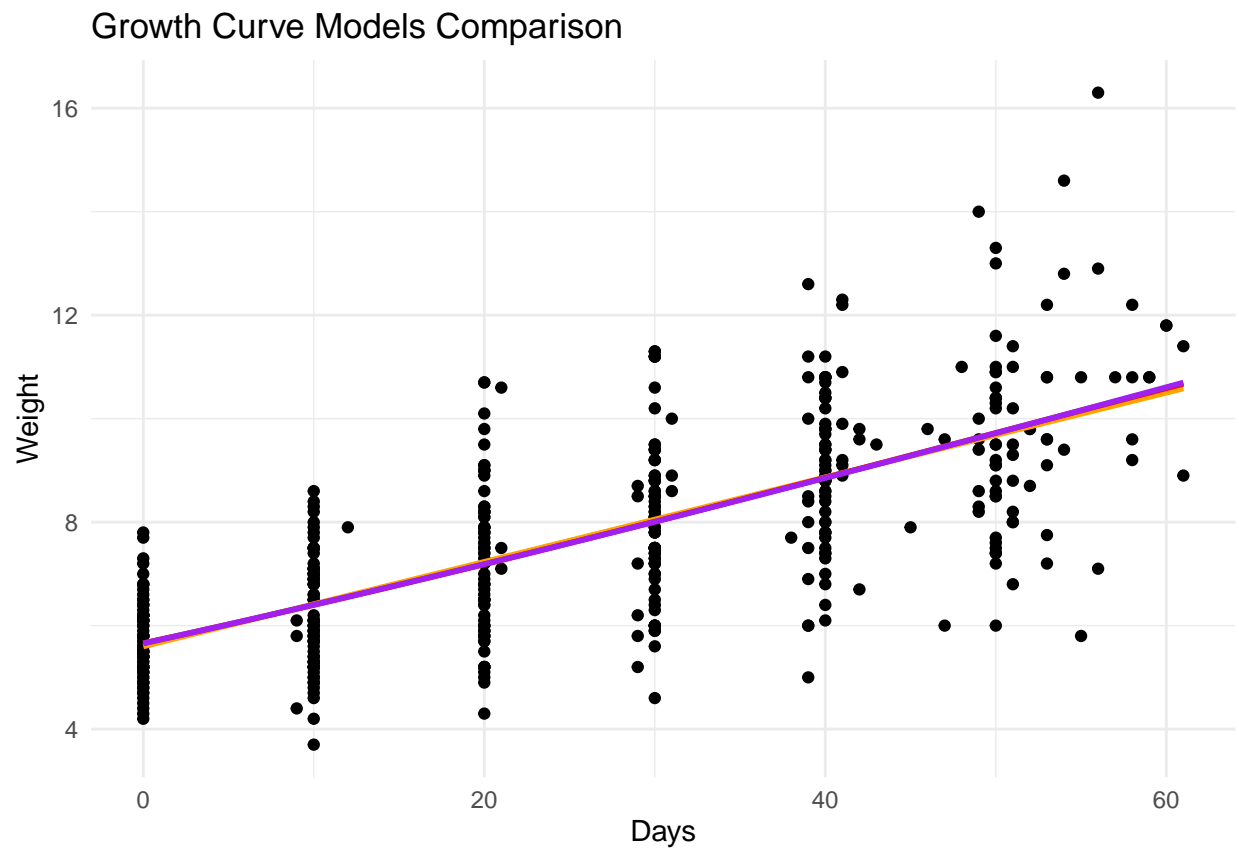
        method.args = list(start = coef(linear.model))) +
geom_smooth(method = "nls", formula = y ~ K / (1 + exp(-r * (x - t))), se = FALSE,
  ↪ color = "red",
        method.args = list(start = coef(logistic.model))) +
geom_smooth(method = "nls", formula = y ~ K * exp(-exp(-r * (x - t))), se = FALSE,
  ↪ color = "purple",
        method.args = list(start = coef(gompertz.model))) +
labs(title = "Growth Curve Models Comparison",
  x = "Days",
  y = "Weight") +
theme_minimal()

```

```

## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).
## Removed 2 rows containing non-finite values (`stat_smooth()`).
## Removed 2 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 2 rows containing missing values (`geom_point()`).

```



```

#clean up
rm(linear.model, logistic.model, gompertz.model, models.1, mod.names.1)

```

The different growth curves all lie within a delta AIC of 2, which means the general difference between the fit of the different growth curves to the data is not strong. Therefore, we proceed with the linear model structure.

Model Now that we have determined the base structure of the repeated measures model, we can add more explanatory variables.

```
#without scaled continuous variables, this will not work
SurvivorsRM_Day60$sMLH_SNP.new_scale <- scale(SurvivorsRM_Day60$sMLH_SNP.new)
SurvivorsRM_Day60$sMLH_SNP.new_mum_scale <- scale(SurvivorsRM_Day60$sMLH_SNP.new_mum)
SurvivorsRM_Day60$froh_scale <- scale(SurvivorsRM_Day60$froh)
SurvivorsRM_Day60$froh_mum_scale <- scale(SurvivorsRM_Day60$froh_mum)

# Calculate the correlation matrix for numeric variables
cor_matrix <- cor(SurvivorsRM_Day60[, c("Age_Days_scale", "sMLH_SNP.new_scale",
  ↪ "sMLH_SNP.new_mum_scale")], use = "pairwise.complete.obs")
eigenvalues <- eigen(cor_matrix)$values
#the eigenvalues are close to 1, which indicate some level of collinearity, but not
  ↪ enough to create a considerable effect

#Rule of thumb: random factors should have >= 5 levels.
#Random vs fixed effects:
#Random effects should be categorical. Interested in controlling for/estimate the
  ↪ variance.
#Fixed effects are the effects we know from linear models

#growth varying per day, both intercept and slope vary per individual including
  ↪ explanatory variables
#model fitted with log transformed weight_kg
RM2_EV_log <- lmer(Weight_kg_log ~ Age_Days_scale +
  sMLH_SNP.new_scale +
  Sex +
  Season +
  Beach +
  sMLH_SNP.new_mum_scale +
  (1 + scale(Age_Days) | ID),
  data = SurvivorsRM_Day60)

#summary(RM2_EV_log)
sjPlot::tab_model(RM2_EV_log)
```

Weight kg log

Predictors

Estimates

CI

p

(Intercept)

1.91

1.84 – 1.99

<0.001

Age Days scale

0.19

0.17 – 0.20
 <0.001
 sMLH SNP new scale
 0.00
 -0.04 – 0.04
 0.952
 Sex [M]
 0.11
 0.03 – 0.19
 0.004
 Season [1920]
 0.04
 -0.02 – 0.10
 0.201
 Beach [SSB]
 0.03
 -0.04 – 0.09
 0.433
 sMLH SNP new mum scale
 -0.04
 -0.07 – -0.00
 0.033
 Random Effects
 2
 0.01
 00 ID
 0.02
 11 ID.scale(Age_Days)
 0.00
 01 ID
 0.34
 ICC
 0.65
 N ID
 73
 Observations

439

Marginal R2 / Conditional R2

0.590 / 0.856

```
#Age, Sex and mums sMLH_SNP has a significant effect

#test model fit
# Predict values for your existing data
predicted_values <- predict(RM2_EV_log, type = "response")
na_values <- rep(NA, 20) #to make the predicted no. of values equal to the observed data
predicted_values <- c(predicted_values, na_values)

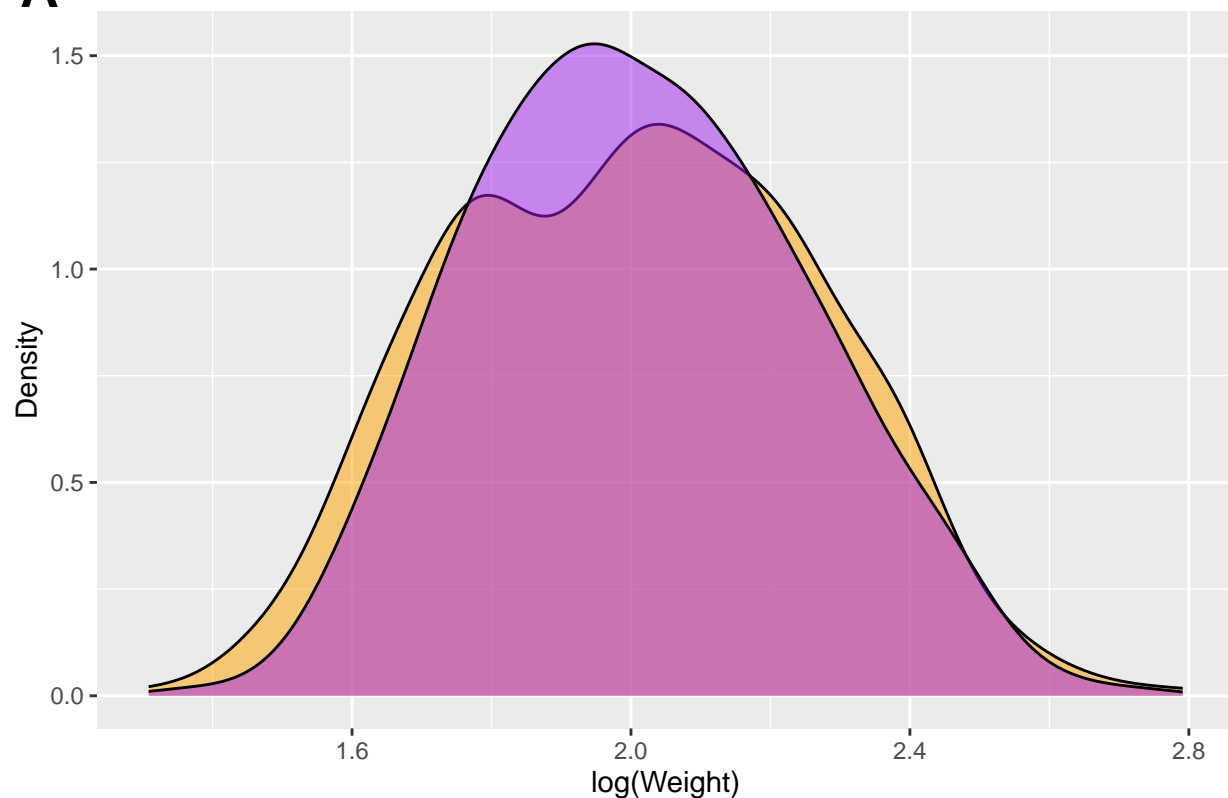
#plot the densities of the predicted and observed data
log_model <- ggplot(SurvivorsRM_Day60, aes(x = Weight_kg_log)) +
  geom_density(fill = "orange", alpha = 0.5) +
  geom_density(aes(x = predicted_values), fill = "purple", alpha = 0.5) +
  labs(x = "log(Weight)", y = "Density") +
  ggtitle("Observed vs. Predicted Weight")

plot_grid(log_model, labels = "AUTO", label_size = 20)
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_density()`).
```

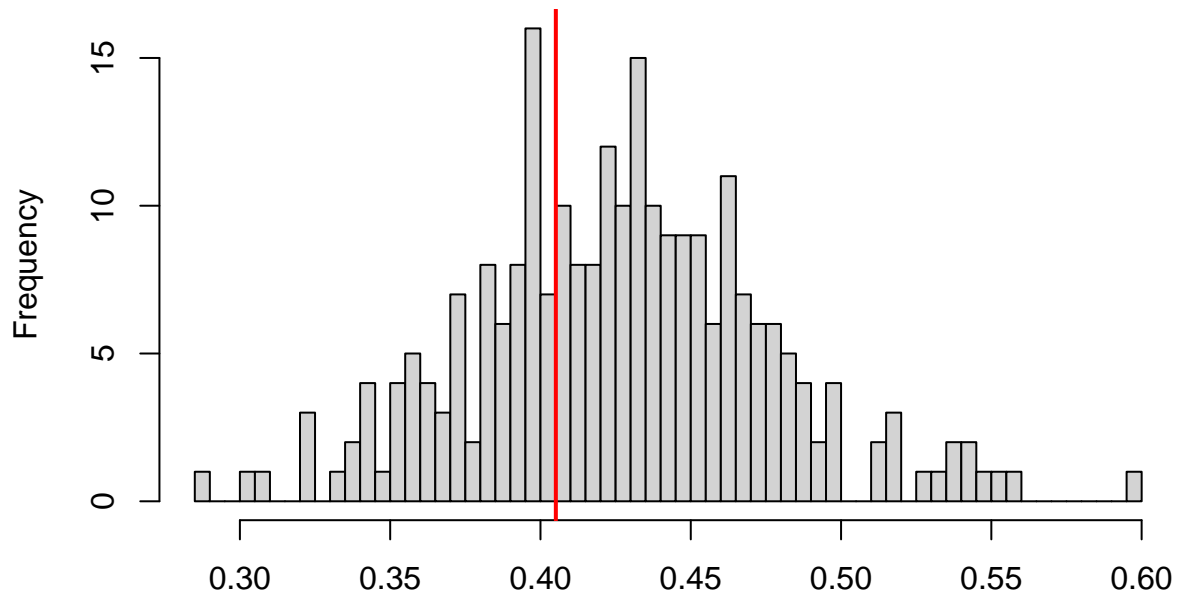
```
## Warning: Removed 20 rows containing non-finite values (`stat_density()`).
```

A Observed vs. Predicted Weight



```
testDispersion(RM2_EV_log) #good fit
```

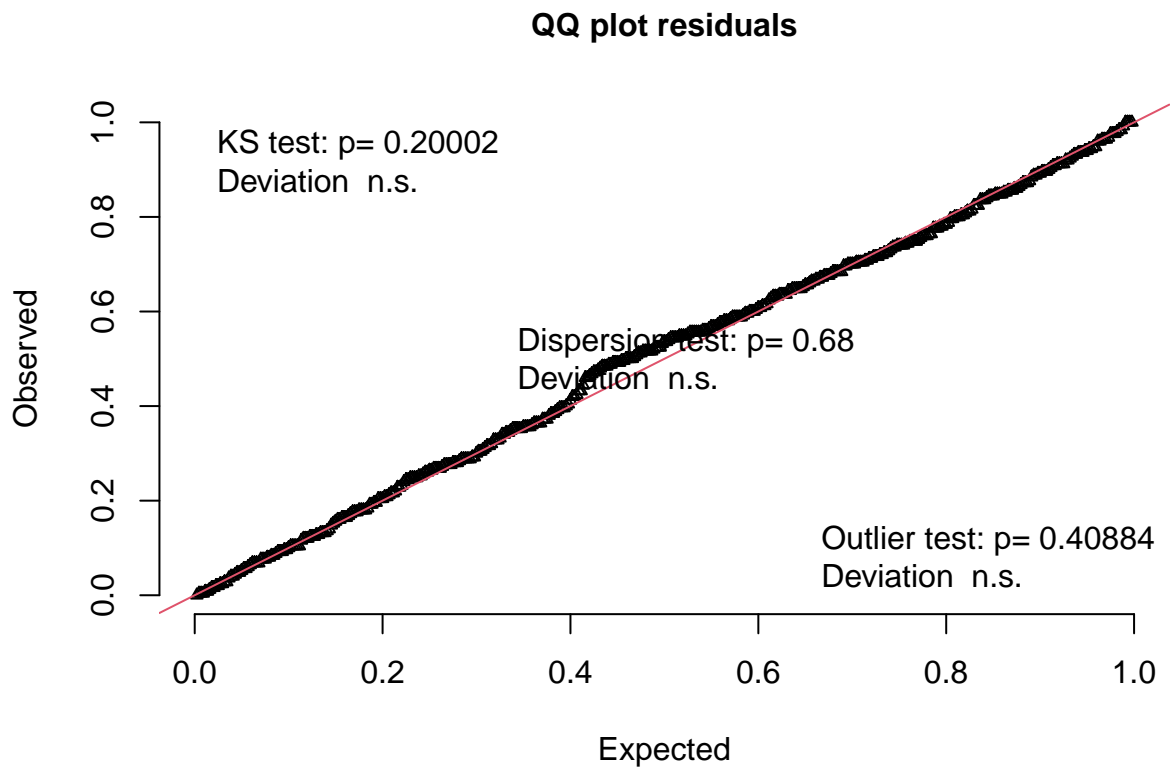
**DHARMa nonparametric dispersion test via sd of
residuals fitted vs. simulated**



Simulated values, red line = fitted model. p-value (two.sided) = 0.68

```
##  
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 0.95002, p-value = 0.68  
## alternative hypothesis: two.sided
```

```
plotQQunif(RM2_EV_log) #good fit
```



```
#growth model with froh
RM2_EV_froh <- lmer(Weight_kg_log ~ froh_scale +
  froh_mum_scale +
  Sex +
  Season +
  Beach +
  Age_Days_scale +
  (1 + scale(Age_Days) | ID),
  data = SurvivorsRM_Day60)
#summary(RM2_EV_froh)
sjPlot::tab_model(RM2_EV_froh)
```

Weight kg log

Predictors

Estimates

CI

p

(Intercept)

1.90

1.83 – 1.97

<0.001

froh scale
 -0.02
 -0.05 – 0.01
 0.226
 froh mum scale
 0.02
 -0.01 – 0.06
 0.165
 Sex [M]
 0.12
 0.05 – 0.19
 <0.001
 Season [1920]
 0.04
 -0.02 – 0.11
 0.193
 Beach [SSB]
 0.03
 -0.03 – 0.09
 0.362
 Age Days scale
 0.19
 0.17 – 0.20
 <0.001
 Random Effects
 2
 0.01
 00 ID
 0.02
 11 ID.scale(Age_Days)
 0.00
 01 ID
 0.30
 ICC
 0.65
 N ID

73

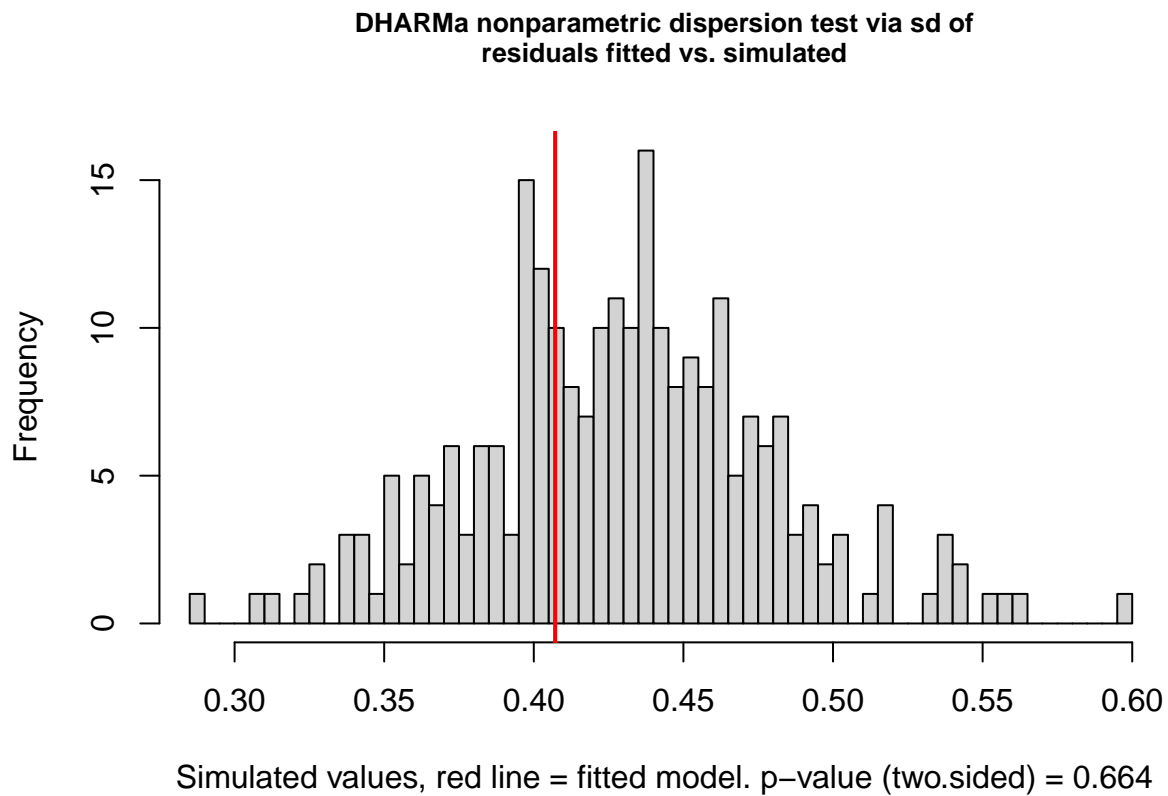
Observations

439

Marginal R2 / Conditional R2

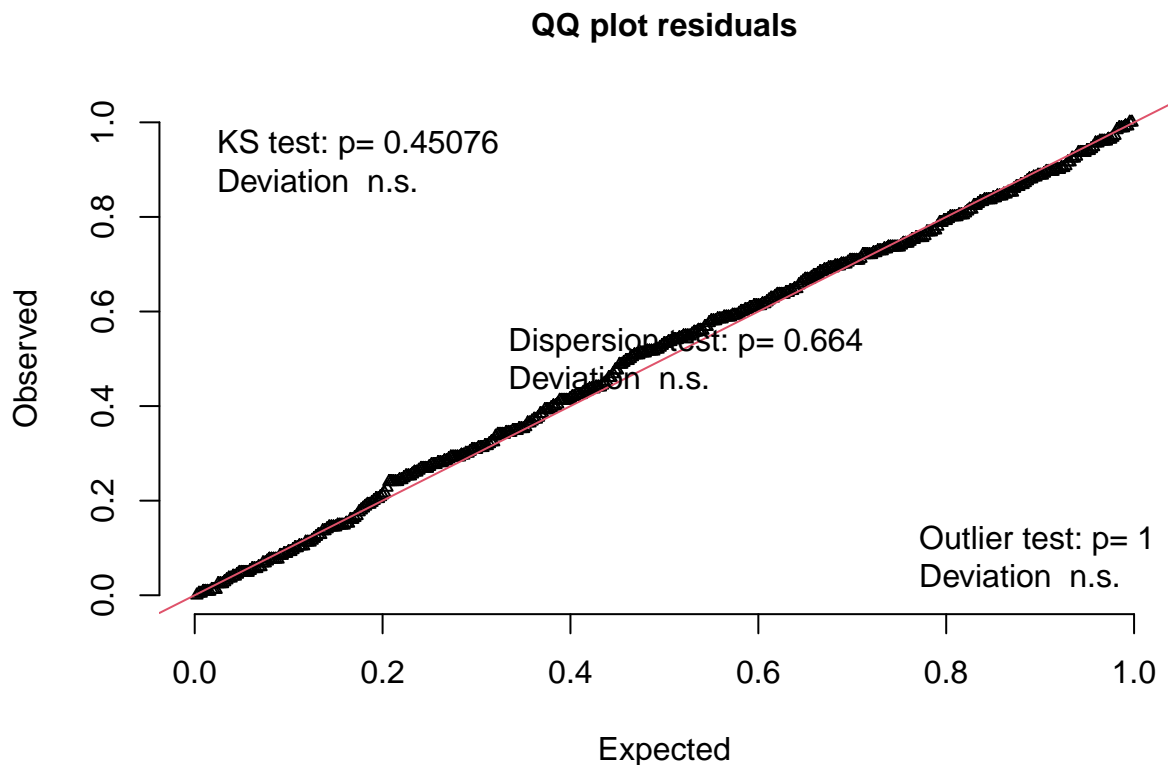
0.587 / 0.856

```
testDispersion(RM2_EV_froh) #good fit
```



```
##  
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 0.9491, p-value = 0.664  
## alternative hypothesis: two.sided
```

```
plotQQunif(RM2_EV_froh) #good fit
```



```
#clean up
rm(cor_matrix, eigenvalues, na_values)
```

Individual growth curves

We can use the repeated measures dataset to build a weight model and extract individual growth curves, that we can further explore. The growth curves estimate trajectories that are unique to each individual.

```
#Get individual growth curves (utilizes the growth model from the previous section)
Coefficients <- coef(RM2_EV_log)[[1]][c("scale(Age_Days)", "(Intercept)")]
setnames(Coefficients, c("Slope", "Intercept"))
Coefficients <- data.table(ID = rownames(Coefficients), Coefficients)

#The model was built on log transformed weight data, so the coefficients have been back
→ transformed.
Coefficients$Slope <- exp(Coefficients$Slope)
Coefficients$Intercept <- exp(Coefficients$Intercept)
UniqueSurvivors_Day60 <- left_join(UniqueSurvivors_Day60, Coefficients, by = 'ID')

#shapiro.test(UniqueSurvivors_Day60$Slope) #normally distributed
#shapiro.test(UniqueSurvivors_Day60$Intercept) #normally distributed

I.model <- lm(Intercept ~ Sex +
              Season +
              Beach +
```

```

        sMLH_SNP.new +
        sMLH_SNP.new_mum,
        data = UniqueSurvivors_Day60)

#plot(I.model)
#summary(I.model) #intercept is essentially birth weight, and is not explained strongly
  ↳ by any of the other parameters
#testDispersion(I.model) #good fit
#plotQQunif(I.model) #deviation not significant

S.model <- lm(Slope ~ Sex +
              Season +
              Beach +
              sMLH_SNP.new +
              sMLH_SNP.new_mum,
              data = UniqueSurvivors_Day60)

#plot(S.model)
#summary(S.model) #slope is explained by sex with males growing faster
#testDispersion(S.model) #good fit
#plotQQunif(S.model) #deviation not significant

#repeat with froh values

#Get individual growth curves (utilizes the growth model from the previous section)
Coefficients.froh <- coef(RM2_EV_froh)[[1]][c("scale(Age_Days)", "(Intercept)")]
setnames(Coefficients.froh, c("Slope.froh", "Intercept.froh"))
Coefficients.froh <- data.table(ID = rownames(Coefficients.froh), Coefficients.froh)

#The model was built on log transformed weight data, so the coefficients have been back
  ↳ transformed.
Coefficients.froh$Slope.froh <- exp(Coefficients.froh$Slope.froh)
Coefficients.froh$Intercept.froh <- exp(Coefficients.froh$Intercept.froh)
UniqueSurvivors_Day60 <- left_join(UniqueSurvivors_Day60, Coefficients.froh, by = 'ID')

#shapiro.test(UniqueSurvivors_Day60$Slope) #normally distributed
#shapiro.test(UniqueSurvivors_Day60$Intercept) #normally distributed

I.model.froh <- lm(Intercept.froh ~ Sex +
                  Season +
                  Beach +
                  froh +
                  froh_mum,
                  data = UniqueSurvivors_Day60)

#plot(I.model.froh)
#summary(I.model.froh) #intercept is essentially birth weight, and is not explained
  ↳ strongly by any of the other parameters
#testDispersion(I.model.froh) #good fit
#plotQQunif(I.model.froh) #deviation not significant

S.model.froh <- lm(Slope.froh ~ Sex +
                  Season +
                  Beach +

```

```
froh +  
froh_mum,  
data = UniqueSurvivors_Day60)
```

```
#plot(S.model.froh)  
#summary(S.model.froh) #slope is explained by sex with males growing faster  
#testDispersion(S.model.froh) #good fit  
#plotQQunif(S.model.froh) #deviation not significant
```

Manuscript figures

Figure 1: map and seasonal data

```
#~~~~~#
# Seasonal data ####
#~~~~~#

seasonal_data <- read.table(here("Data", "Raw", "seasonal_data.txt"), sep = "\t",
  ↪ stringsAsFactors = F, header = T)

#~~ make figure

# Use Martin Stoffel's GGplot theme as a base
source(here("Rcode", "anneke_theme.R"))

# Make a list for the theme so it is the same for all figures
gglayer_theme <- list(
  scale_x_discrete(labels = c(`2017-2018` = "2018", `2018-2019` = "2019", `2019-2020` =
    ↪ "2020", `2020-2021` = "2021")),
  theme_anneke(),
  theme(axis.line.x = element_line(colour = 'black', linetype='solid'),
    axis.line.y = element_line(colour = 'black', linetype='solid'),
    plot.title = element_text(size = rel(1)))
)

# Make sub plots

# Plot a is blank canvas + title where later the map gets added
p_a <- ggplot(seasonal_data %>% filter(variable=="SSB ESTIMATED NUMBER OF FEMALE
  ↪ BREEDERS"),
  aes(x = season, y = mean)) +
  #geom_pointrange(aes(ymin = CI95_low, ymax = CI95_high)) +
  #geom_point(shape = 22, size = 4, fill = "#eb7f86") +
  labs(title="(a) Map of Bird Island", x= "", y="") +
  gglayer_theme +
  theme(axis.line.x = element_blank(),
    axis.line.y = element_line(colour = 'white', linetype='solid'),
    axis.text = element_text(colour = "white"),
    panel.grid=element_blank(),
    panel.grid.major=element_blank(),
    panel.grid.minor=element_blank())

p_breeders <- ggplot(seasonal_data%>% filter(variable=="SSB ESTIMATED NUMBER OF FEMALE
  ↪ BREEDERS"),
  aes(x = season, y = mean)) +
  geom_pointrange(aes(ymin = CI95_low, ymax = CI95_high)) +
  geom_point(shape = 22, size = 4, fill = "#ea4f88") +
  labs(title="(b) Female breeders", x= "Year", y="No. of breeders") +
  gglayer_theme

p_bm <- ggplot(seasonal_data%>% filter(variable=="SSB FEMALE PUP BIRTH MASS (kg)"),
```

```

      aes(x = season, y = mean)) +
    geom_pointrange(aes(ymin = CI95_low, ymax = CI95_high)) +
    geom_point(shape = 22, size = 4, fill = "#4b2991") +
    labs(title="(c) Female pup birth mass", x= "Year", y="Birth mass (kg)") +
    gglayer_theme

p_foraging <- ggplot(seasonal_data %>% filter(variable=="FWB FEMALE FORAGING TRIP
  ↳ DURATION (days)"),
      aes(x = season, y = mean)) +
    geom_pointrange(aes(ymin = CI95_low, ymax = CI95_high)) +
    geom_point(shape = 22, size = 4, fill = "#f6a97a") +
    labs(title="(d) Female foraging trip duration", x="Year", y="Time at sea (days)") +
    gglayer_theme

```

```

#####
# Bird Island map ####
#####

```

```
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
library(chron)
```

```
library(rgdal)
```

```
## Loading required package: sp
```

```
## Warning: package 'sp' was built under R version 4.0.5
```

```
## Please note that rgdal will be retired during October 2023,
```

```
## plan transition to sf/stars/terra functions using GDAL and PROJ
```

```
## at your earliest convenience.
```

```
## See https://r-spatial.org/r/2023/05/15/evolution4.html and https://github.com/r-spatial/evolution
```

```
## rgdal: version: 1.6-7, (SVN revision 1203)
```

```
## Geospatial Data Abstraction Library extensions to R successfully loaded
```

```
## Loaded GDAL runtime: GDAL 3.4.1, released 2021/12/27
```

```
## Path to GDAL shared files: C:/Users/localadmin/Documents/R/win-library/4.0/rgdal/gdal
```

```
## GDAL binary built with GEOS: TRUE
```

```
## Loaded PROJ runtime: Rel. 7.2.1, January 1st, 2021, [PJ_VERSION: 721]
```

```
## Path to PROJ shared files: C:/Users/localadmin/Documents/R/win-library/4.0/rgdal/proj
```

```
## PROJ CDN enabled: FALSE
```

```
## Linking to sp version:1.4-5
```

```
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
```

```
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
```

```
library(ggspatial)
```

```
library(ggsn)
```

```
## Warning: package 'ggsn' was built under R version 4.0.5
```

```
## Loading required package: grid
```

```
# bird island maps
```

```
bi_coast <- st_read(here("Rcode", "Bird Island Map", "Map_Old",
```

```
  ↳ "BI_Coast_Projected_new.shp"))
```

```

## Reading layer `BI_Coast_Projected_new' from data source
##   `C:\Uni\10_Growth_msats-2017-2020\Rcode\Bird Island Map\Map_Old\BI_Coast_Projected_new.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 28 features and 5 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:   xmin: -38.08212 ymin: -54.02197 xmax: -38.00766 ymax: -53.99703
## Geodetic CRS:   WGS 84

bi_r <- st_read(here("Rcode", "Bird Island Map", "rivers_lines", "sg_bird_rivers.shp"))
  ↪ #rivers

## Reading layer `sg_bird_rivers' from data source
##   `C:\Uni\10_Growth_msats-2017-2020\Rcode\Bird Island Map\rivers_lines\sg_bird_rivers.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 23 features and 9 fields
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:   xmin: -70432.32 ymin: 108837.5 xmax: -66966.1 ymax: 110309
## Projected CRS:  WGS 84 / South Georgia Lambert

bi_c <- st_read(here("Rcode", "Bird Island Map", "contours", "sg_bird_contours.shp")) #
  ↪ contours

## Reading layer `sg_bird_contours' from data source
##   `C:\Uni\10_Growth_msats-2017-2020\Rcode\Bird Island Map\contours\sg_bird_contours.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 842 features and 4 fields
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:   xmin: -70879.52 ymin: 108350.2 xmax: -66120.84 ymax: 110974.7
## Projected CRS:  WGS 84 / South Georgia Lambert

bi <- st_read(here("Rcode", "Bird Island Map", "coastline", "sg_bird_coast.shp"))
  ↪ #surface

## Reading layer `sg_bird_coast' from data source
##   `C:\Uni\10_Growth_msats-2017-2020\Rcode\Bird Island Map\coastline\sg_bird_coast.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 122 features and 7 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:   xmin: -71251.06 ymin: 107969.6 xmax: -65945.78 ymax: 111316
## Projected CRS:  WGS 84 / South Georgia Lambert

# outline of ssb and fwb, made in Google Earth
ssb <- st_read(here("Rcode", "Bird Island Map", "beachs", "SSB.kml", "doc.kml"))

## Reading layer `SSB.kmz' from data source
##   `C:\Uni\10_Growth_msats-2017-2020\Rcode\Bird Island Map\beachs\SSB.kml\doc.kml'
##   using driver `KML'
## Simple feature collection with 1 feature and 2 fields
## Geometry type: POLYGON
## Dimension:      XYZ

```



```

## Bounding box:  xmin: -38.05125 ymin: -54.01178 xmax: -38.05022 ymax: -54.01133
## z_range:      zmin: 0 zmax: 0
## Geodetic CRS:  WGS 84

fwb <- st_read(here("Rcode", "Bird Island Map", "beachs", "FWB.kml", "doc.kml"))

## Reading layer `FWB.kmz' from data source
## `C:\Uni\10_Growth_msats-2017-2020\Rcode\Bird Island Map\beachs\FWB.kml\doc.kml'
## using driver `KML'
## Simple feature collection with 1 feature and 2 fields
## Geometry type: POLYGON
## Dimension:      XYZ
## Bounding box:   xmin: -38.05219 ymin: -54.00916 xmax: -38.05081 ymax: -54.00836
## z_range:        zmin: 0 zmax: 0
## Geodetic CRS:   WGS 84

#### mapping bird island ----
plot.bi.color <- ggplot() +
  geom_sf(data = bi_coast, fill = "#ADADAD") +
  #geom_sf(data = bi_r, color = "blue") +
  #geom_sf(data = bi_c) +
  geom_sf(data = ssb, fill = "#872ca2") + #ssb
  geom_sf(data = fwb, fill = "#fa7876") + #fwb
  theme(legend.position = "none") +
  theme_void()
#+ scalebar(data = bi_coast, dist = 500, dist_unit = "m",
# transform = TRUE, model = "WGS84",
# location = "bottomright",
# box.fill = "white", border.size = 0.5))

#### mapping study colonies ----
#adds box around study colonies on bird island map
plot.bi.color. <- plot.bi.color +
  annotate(geom = "rect",
    xmin = -38.060,
    xmax = -38.045,
    ymin = -54.014,
    ymax = -54.0065,
    fill = NA, # transparent bg
    color = "black" )

plot.bi.beaches.color <- ggplot() +
  geom_sf(data = bi_coast, fill = "NA") +
  geom_sf(data = bi_coast, fill = "#D4CEC2") + # "#eaeaea"
  geom_sf(data = fwb, fill = "#fa7876") +
  geom_sf(data = ssb, fill = "#872ca2") +
  theme(legend.position = "none") +
  theme_void()

#add text
plot.bi.beaches.color <- plot.bi.beaches.color +
  coord_sf(xlim = c(-38.060, -38.045),
    ylim = c(-54.014, -54.0065),
    expand = FALSE) +

```

```

annotation_scale(aes(location="br", style = "ticks")) +
theme(panel.border = element_rect(colour = "black", fill=NA, size=1)) +
annotate(geom = "text",
          x = -38.05,
          y = -54.0092,
          label = "FWB",
          color = "#fa7876",
          fontface = "bold") +
annotate(geom = "text",
          x = -38.05,
          y = -54.011,
          label = "SSB",
          color = "#872ca2",
          fontface = "bold") #+

```

```

## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

# annotate(geom = "text",
#           x = -38.058,
#           y = -54.0068,
#           label = "Bird Island",
#           color = "black",
#           fontface = "bold") )

#combine
map <- ggdraw(p_a) + # empty canvas with title to match other plots
  draw_plot(plot.bi.beaches.color, x= 0.07, scale = .8) + # study colonies
  draw_plot(plot.bi.color., 0.07, .48, .5, .5, scale = 1.3) #+ # Bird Iland
  #theme(plot.margin = margin(1,1,1,1.2, "cm"))

#map

#~~~~~#
# Final plot ####
#~~~~~#

P_seasonal <- plot_grid(map, p_breeders, p_bm, p_foraging)
# , labels = c("(a) Map of Bird Island",
#               "(b) Female breeders",
#               "(c) Female pup birth mass",
#               "(d) Female foraging trip duration"),
# label_size = 12, label_fontface = "plain")

P_seasonal

```

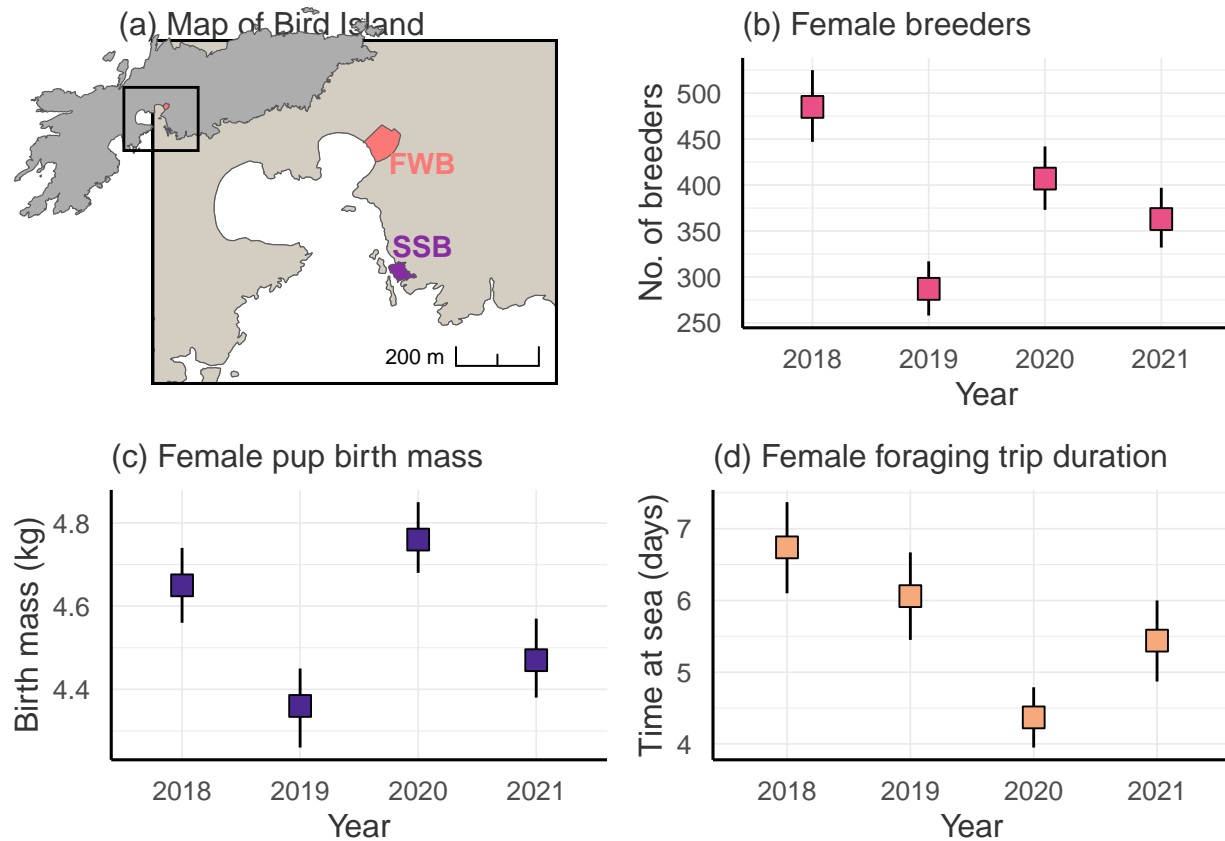


Figure 2: forest plots microsatellite models

```
#~~~~~#
# Make general theme #####
#~~~~~#

# Color non sig effects
col1 = "dimgrey"
# Color sig effects
col2 = "#fa7876" "#ea4f88" # "#872ca2"

# Use Martin Stoffel's GGplot theme as a base
source(here("Rcode", "anneke_theme.R"))

# Make a list for the theme so it is the same for all figures
gglayer_theme <- list(
  geom_point(shape = 22, size = 3, fill = "black"),
  theme_anneke(),
  theme(axis.line.x = element_line(colour = 'black', linetype='solid'),
        axis.line.y = element_line(colour = 'black', linetype='solid'),
        axis.text.y = element_text(colour = 'black'),
        plot.title = element_text(size = rel(1)))
)
```

```

plot_label <- c(
  `(Intercept)` = "Intercept",
  sMLH_msat39_pup = "pup sMLH",
  Pup_SexM = "pup sex [M]",
  Pup_BirthWeight = "pup birth mass",
  sMLH_msat39_mum = "mother sMLH",
  Mum_Age = "mother age",
  Year2018 = "year [2019]",
  Year2019 = "year [2020]",
  Year2020 = "year [2021]")

#~~~~~#
# Function to create forest plots ####
#~~~~~#

plot_data_models = function(data_model, p_title) {

  # #Test function
  # p_title <- "(a) Birth mass"
  # data_model <- m1birthmass

  p <- plot_model(data_model,
    show.intercept = F,
    type="est",
    #order.terms = c(1, 4, 2, 3)
    ci.lvl=0.95,
    axis.labels = plot_label,
    title=p_title,
    colors = "black",
    line.size=0.5,
    vline.color = "#cccccc") # Look into theme_sjplot to make pretty #,
    ↪ sort.est = TRUE

  p <- p + gglayer_theme

  #p

  # Adjust line size vertical line
  p$layers[[1]]$aes_params$size <- 1.3

  #~~ Possible sjPlot bug, but in the binomial model, the whiskers for one of the fixed
  ↪ effects are missing. Fix this:

  if(class(data_model)[1]=="glm") {
    if(data_model$family[1]=="binomial") {

      p <- p + scale_y_log10(labels = scales::math_format(expr = .x))

      q <- ggplot_build(p)

      # In q$data[[3]], y = the model estimate, and ymin and ymax are the 95% CI (check
      ↪ with: confint(m1birthmass, level = 0.95) ) for LM

```

```

# For the binomial model it is the transferred Odds ratio, which is done for the
  ↪ figure.
# Therefore the values do NOT match the odds ratios (which one can calculate with
  ↪ exp(cbind(Odds_Ratio = confint(m1survival))) for example).
# Instead the values are log10 transformed. So they match with
  ↪ log10(exp(cbind(Odds_Ratio = confint(m1survival))))

# Replace ymin and ymax with correct values

q$data[[3]][["ymin"]] <- log10(exp(cbind(Odds_Ratio = confint(m1survival))))[-1,1]
q$data[[3]][["ymax"]] <- log10(exp(cbind(Odds_Ratio = confint(m1survival))))[-1,2]

}
} else {
  p <- p + scale_y_continuous(labels = scales::label_number())

  q <- ggplot_build(p) }

#~~ Identify significant variables and give them a different color

q$data[[3]] = q$data[[3]] %>%
  mutate(fill = ifelse(ymin < 0 & ymax > 0, col1, col2))

#q$data[[2]]$colour = q$data[[3]]$fill # only necessary if you keep the dots instead of
  ↪ the squares

# Line below to remove the dots complete. Gives a warning for missing values, but that
  ↪ is not a problem
q$data[[2]]$colour = NA

q$data[[4]]$fill = q$data[[3]]$fill

final_plot <- ggplot_gtable(q)

#plot(final_plot)

return(final_plot)
}

#~~ Apply function to 3 models
# nb warnings are because I am removing dots and adding squares in the function above

p.bw <- plot_data_models(m1birthmass, "(a) Birth mass")

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Warning: Removed 7 rows containing missing values (`geom_point()`).

```

```
p.surv <- plot_data_models(m1survival, "(b) Survival")
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Waiting for profiling to be done...
##
## Waiting for profiling to be done...
## Warning: Removed 8 rows containing missing values (`geom_point()`).
```

```
p.wg <- plot_data_models(m1growth, "(c) Growth")
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Warning: Removed 8 rows containing missing values (`geom_point()`).
```

```
#plot(p.wg)
```

```
##~ Save plots
```

```
all_plots <- cowplot::plot_grid(p.bw, p.surv, p.wg,
                                nrow = 1)
```

```
#align = "v"
```

```
all_plots
```

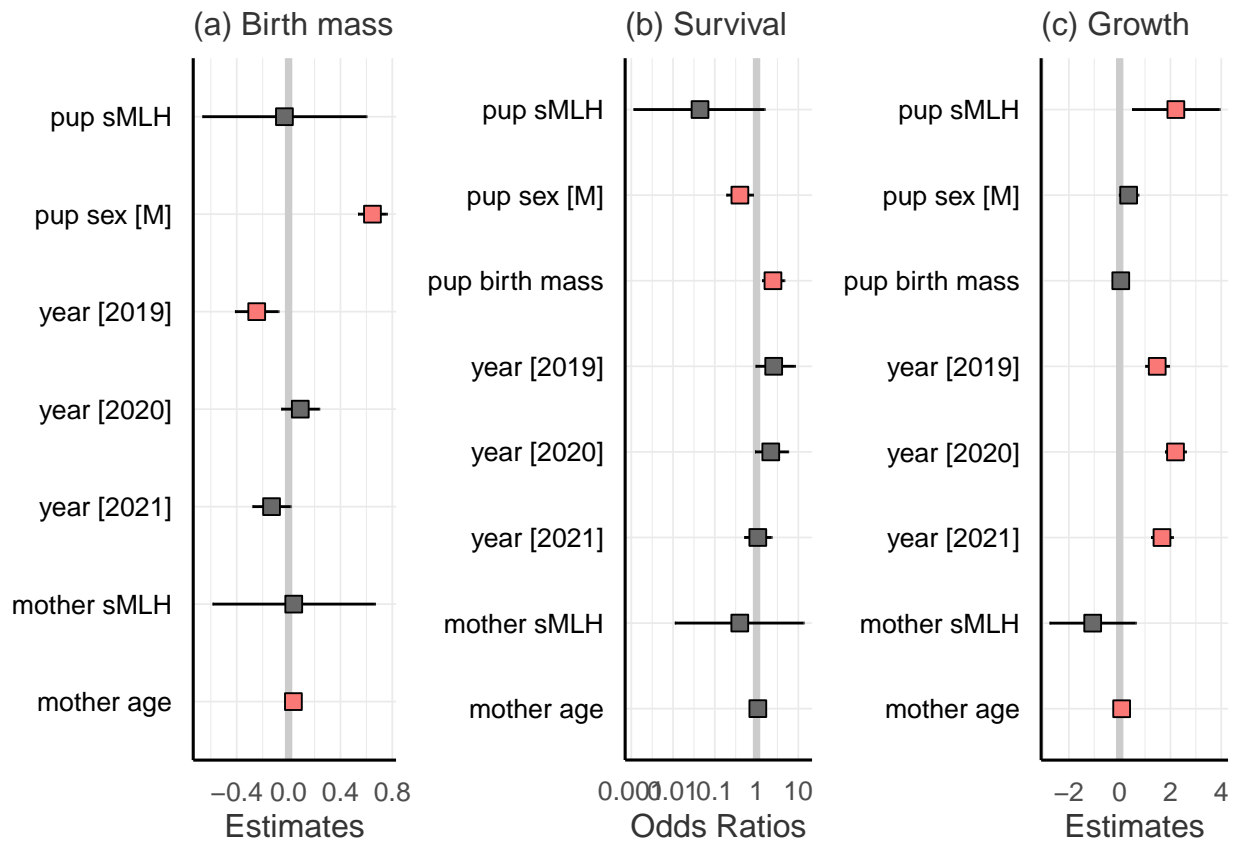


Figure 3: forest plots SNP models

Forest plots for birth weight, survival and growth models

Plots for figure

```
AllCurves <- ggplot(data = SurvivorsRM_Day60, aes(x = Age_Days, y = Weight_kg, group =
  ID)) +
  geom_smooth(method = "lm", se = F, colour = "grey", linewidth = 0.8, alpha = 0.5) +
  geom_point(shape = 0, colour = "black", size = 1) +
  geom_abline(slope = 0.081732, intercept = 5.597941, colour = "black", linewidth = 1.1)
  #average
  #geom_abline(slope = 0.080226, intercept = 5.502678, colour = "orange", size = 2) +
  #annotate(geom="text", y = 15.5, x = 15, size = 5, label = "FWB:  $y = 0.080226 \cdot \text{age} + 5.502678$ ", color = "orange") +
  #geom_abline(slope = 0.080821, intercept = 5.511745, colour = "purple", size = 2) +
  #annotate(geom="text", y = 16, x = 15, size = 5, label = "SSB:  $y = 0.080821 \cdot \text{age} + 5.511745$ ", color = "purple") +
  theme_bw(base_size = 18) + #removes background color
  theme(panel.border = element_blank()) + #removes border lines
  theme(axis.line = element_line(colour = "black")) + #adds in axis lines
  xlab("Age (Days)") + #name of x lab
  ylab("Weight (Kg)") + #name of y lab
  ggtitle("(b) Linear growth curves incl. average") #title of plot

#https://quantdev.ssri.psu.edu/tutorials/growth-modeling-basics

p.allcurves <- AllCurves +
  gglayer_theme_alt2 +
  theme(axis.title.y=element_text(angle=0, vjust = 0.5))

#Plot for poster with illustrative examples
PlotIllu <- subset(SurvivorsRM_Day60, ID %in% c('H14', 'H1', 'C12', 'T4', 'H7'))

SubIllu <- ggplot(data = PlotIllu, aes(x = Age_Days, y = Weight_kg, colour = ID)) +
  geom_line(linetype = "dashed", size = 1, alpha = 0.8) +
  geom_smooth(method = "lm", se = F, size = 1.2, alpha = 1) +
  #scale_color_brewer(palette="PuOr") +
  scale_color_carto_d(palette = "ag_Sunset") + #colorblind friendly palette
  theme_bw(base_size = 18) + #removes background color
  theme(panel.border = element_blank()) + #removes border lines
  theme(axis.line = element_line(colour = "black")) + #adds in axis lines
  theme(legend.position = c(0.15, 0.92)) +
  theme(legend.background = element_rect(fill="NA")) +
  xlab("Age (Days)") + #name of x lab
  xlim(0,60) + #x axis limits
  ylab("Weight (Kg)") + #name of y lab
  ylim(3.5,16.4) +
  guides(fill=guide_legend(title="ID")) + #name of legend
  ggtitle("(a) Individual growth curves of 5 pups") #title of plot
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.

i Please use `linewidth` instead.

This warning is displayed once every 8 hours.

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

p.IDGrowth <- SubIllu + gglayer_theme_alt + theme(legend.position = c(0.12, 0.73)) +
  #theme(legend.background = element_rect(fill="NA")) +
  guides(fill=guide_legend(title="ID")) +
  theme(axis.title.y=element_text(angle=0, vjust = 0.5))

#png(file = here("Growthplot.png"), # The directory you want to save the file in
# width = 100, # The width of the plot in inches
# height = 50)

#plot_grid(SubIllu, AllCurves, labels = "AUTO", label_size = 20)

library(ggsignif)

## Warning: package 'ggsignif' was built under R version 4.0.5

I.sex <- ggplot(data = UniqueSurvivors_Day60, aes (x=Sex, y=Slope, fill = Sex)) +
  geom_boxplot(fill = c("#872ca2", "#ea4f88"), alpha = 0.9) +
  geom_signif(comparisons = list(c("M", "F")), map_signif_level=TRUE, textsize = 7) +
  ylim(0.91, 1.1) +
  theme_anneke() +
  theme(axis.title.y=element_text(angle=0, vjust = 0.5)) +
  ggtitle("(f) Sex driven difference in slope") +
  theme(axis.title = element_text(vjust = 1, hjust = 0.5)) + # angle of axis title
  theme(axis.line.x = element_line(colour = 'black', linetype='solid'),
        axis.line.y = element_line(colour = 'black', linetype='solid'),
        axis.text.y = element_text(colour = 'black'),
        plot.title = element_text(size = rel(1)))

I.sex.froh <- ggplot(data = UniqueSurvivors_Day60, aes (x=Sex, y=Slope.froh, fill = Sex))
  +
  geom_boxplot(fill = c("#872ca2", "#ea4f88"), alpha = 0.9) +
  geom_signif(comparisons = list(c("M", "F")), map_signif_level=TRUE, textsize = 7) +
  ylim(0.91, 1.1) +
  ylab("Slope") + #name of y lab
  theme_anneke() +
  theme(axis.title.y=element_text(angle=0, vjust = 0.5)) +
  ggtitle("(f) Sex driven difference in slope") +
  theme(axis.title = element_text(vjust = 1, hjust = 0.5)) + # angle of axis title
  theme(axis.line.x = element_line(colour = 'black', linetype='solid'),
        axis.line.y = element_line(colour = 'black', linetype='solid'),
        axis.text.y = element_text(colour = 'black'),
        plot.title = element_text(size = rel(1)))

#dev.off()
```

Final figure

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

## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 2 rows containing missing values (`geom_point()`).

## Removed 2 rows containing missing values (`geom_point()`).
```



```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 3 rows containing non-finite values (`stat_boxplot()`).
## Warning: Removed 3 rows containing non-finite values (`stat_signif()`).
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 2 rows containing missing values (`geom_point()`).
## Removed 2 rows containing missing values (`geom_point()`).
## Warning: Removed 3 rows containing non-finite values (`stat_boxplot()`).
## Warning: Removed 3 rows containing non-finite values (`stat_signif()`).
```

