

R-code for ‘Evidence for an Allee effect in a declining fur seal population’

Compiled by R. Nagel

Contents

Download packages and libraries	2
Exploring the data	2
Seasonal differences	3
Attendance	3
Pup birth weight & no. of breeders	7
Colony differences	11
Pup mortality	11
Local density	13
Pups	14
Weight	14
Condition index	20
Mothers	30
Condition index	30
Additional quality traits: weight, length, girth, span	36

This document provides all the R code for the manuscript titled *Evidence for an Allee effect in a declining fur seal population* by Rebecca Nagel, Claire Stainfield, Cameron Fox-Clarke, Camille Toscani, Jaume Forcada and Joseph I. Hoffman. Both the R Markdown file and the data can be downloaded via DRYAD (<https://doi.org/10.5061/dryad.zcrjdfnb0>). If you have any questions, don't hesitate to contact Rebecca Nagel (renagel2@gmail.com).

The data originates from samples of the Antarctic fur seal (*Arctocephalus gazella*) collected on Bird Island, South Georgia between 2018-2020.

Download packages and libraries

In order to repeat analyses presented in this manuscript a number of packages that extend the functionalities of base R are required. These can be installed using the code `install.packages("xxPACKAGENAMExx", dependencies = TRUE)`

```
library(dplyr)
library(rcompanion)
library(tidyverse)
library(lattice)
library(ggplot2)
library(ggpubr)
library(waffle)
library(car)
library(lmerTest)
library(lsmeans)
library(vcd)
library(sjPlot)
library(bbmle)

quantiles_95 <- function(x) {
  r <- quantile(x, probs = c(0.05, 0.25, 0.5, 0.75, 0.95))
  names(r) <- c("ymin", "lower", "middle", "upper", "ymax")
  r
}
```

Exploring the data

```
attendance <- rbind(read.table("attendance_summary_2019.csv", header = TRUE, sep = ","),
  read.table("attendance_summary_2020.csv", header = TRUE, sep = ","))
id <- read.table("MumIDs.csv", header = TRUE, sep = ",")
id <- id %>% select(seal.id, PupDeath)
attendance <- left_join(attendance, id, by = "seal.id")

pup <- read.csv("Sampling_Pup_2018-2020.csv", header = TRUE, sep = ",")
pup$Date <- as.Date(pup$Date, format = "%d-%m-%y")
pup$Season <- as.factor(pup$Season)
```

```
# sex ratio at SSB
knitr::kable(table(pup[pup$Day == 0 & pup$Beach == "SSB", ]$Sex))
```

Var1	Freq
F	29
M	21

```
# sex ratio at FWB
knitr::kable(table(pup[pup$Day == 0 & pup$Beach == "FWB", ]$Sex))
```

Var1	Freq
F	20
M	30

```
mum <- read.csv("Sampling_Mum_2018-2020.csv", header = TRUE, sep = ",")
# take average of girth, span and length measurements
mum_avg <- aggregate(mum[, c("Girth", "Span", "Length")], by = list(mum$ID_Mum), mean,
  na.rm = TRUE)
names(mum_avg)[names(mum_avg) == "Group.1"] <- "ID_Mum"
mum <- inner_join(mum[mum$Day == 0, c("ID_Mum", "Weight", "CI_birthing", "Date", "Beach",
  "Season")], mum_avg, by = "ID_Mum")
rm(mum_avg)
names(mum)[names(mum) == "Weight"] <- "Weight_Birth"

mum$Date <- as.Date(mum$Date, format = "%d-%m-%y")
mum$Season <- as.factor(mum$Season)
```

```
pup <- inner_join(mum[c("ID_Mum", "CI_birthing", "Weight_Birth")], pup, by = "ID_Mum")

names(pup)[names(pup) == "CI_birthing"] <- "CI_MumAvg"
names(pup)[names(pup) == "Weight_Birth"] <- "Weight_MumBirth"
```

Seasonal differences

Attendance

As part of the BAS long-term monitoring program for the Convention for the Conservation of Antarctic Marine Living Resources (CCAMLR), the foraging behavior of focal females fitted with VHF transmitters at FWB (n = 50; 25 per season) was monitored using a fixed-position radio antenna (Televilt RX900) and daily visual checks of the island using a hand-held VHF receiver. Absence or presence of females ashore was noted from first capture, just after giving birth, to final measurement, at pup death or weaning.

Here, we are interested in finding out if there is a significant difference between the maternal time spent at sea (days) between seasons.

```
with(attendance, shapiro.test(time.at.sea.days[season ==
  "2018-2019"]))
```

```
##
## Shapiro-Wilk normality test
##
## data:  time.at.sea.days[season == "2018-2019"]
## W = 0.94492, p-value = 0.0002909
```

```
with(attendance, shapiro.test(time.at.sea.days[season ==
  "2019-2020"]))
```

```
##
## Shapiro-Wilk normality test
##
## data:  time.at.sea.days[season == "2019-2020"]
## W = 0.63519, p-value < 2.2e-16
```

```

# for the first foraging trip
wilcox.test(time.at.sea.days ~ season, data =
  attendance[attendance$trip.number == 1, ], alternative =
  "greater", exact = FALSE)

##
## Wilcoxon rank sum test with continuity correction
##
## data: time.at.sea.days by season
## W = 358, p-value = 7.221e-06
## alternative hypothesis: true location shift is greater than 0

# for all foraging trips
wilcox.test(time.at.sea.days ~ season, data = attendance, alternative
  = "greater", exact = FALSE)

##
## Wilcoxon rank sum test with continuity correction
##
## data: time.at.sea.days by season
## W = 13080, p-value = 5.361e-07
## alternative hypothesis: true location shift is greater than 0

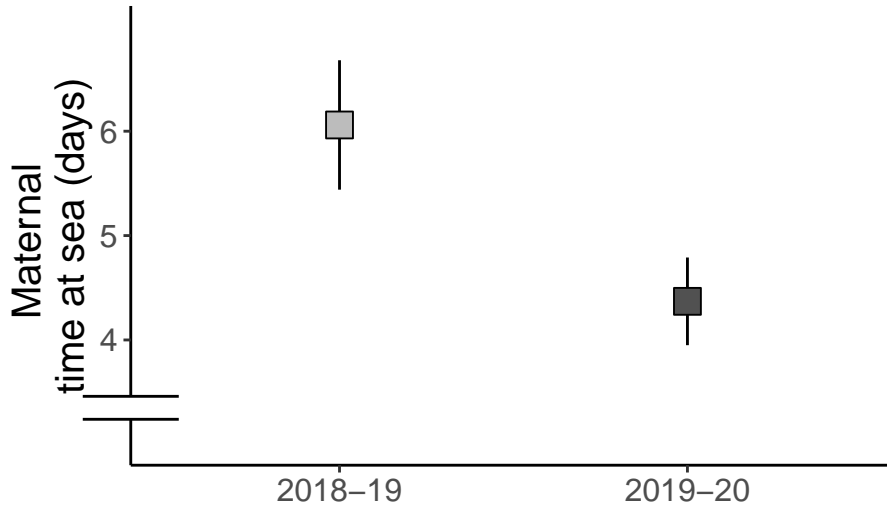
# plotting
attend_av <- groupwiseMean(time.at.sea.days ~ season, data = attendance, conf = 0.95,
  digits = 3)

attend.mum.fwb <- ggplot(data = attend_av, aes(x = season, y = Mean, group = 1, fill =
  season)) + geom_pointrange(aes(ymin = Trad.lower, ymax = Trad.upper)) + #
  geom_line(linetype = 'dashed') + geom_point(shape = 22, size = 5) +
  scale_fill_manual(values = c("#bcbcbc", "#515151")) + scale_x_discrete(labels =
  c(`2018-2019` = "2018-19", `2019-2020` = "2019-20")) + theme_classic() +
  scale_y_continuous(limits = c(3, 7), breaks = c(4, 5, 6)) + labs(y = "Maternal \ntime
  at sea (days)", x = "", title = "(D) Foraging trip duration", subtitle = "") +
  theme(text = element_text(size = 15), legend.position = "none", plot.margin =
  unit(c(10, 5, 5, 5), "mm"), plot.title = element_text(margin = margin(b = 0), size =
  15), plot.subtitle = element_text(hjust = 0.28, margin = margin(t = 5, b = 10), size
  = 12))

# add a break in y-axis
attend.mum.fwb <- ggplotGrob(attend.mum.fwb)
is_yaxis <- which(attend.mum.fwb$layout$name == "axis-l")
yaxis <- attend.mum.fwb$grobs[[is_yaxis]]
yline <- yaxis$children[[1]]
yline$x <- unit(rep(1, 4), "npc")
yline$y <- unit(c(0, 0.1, 1, 0.15), "npc")
yline$id <- c(1, 1, 2, 2)
yline$arrow <- arrow(angle = 90)
yaxis$children[[1]] <- yline
attend.mum.fwb$grobs[[is_yaxis]] <- yaxis
grid.newpage()
grid.draw(attend.mum.fwb)

```

(D) Foraging trip duration



Here, we are looking at the foraging patterns of each individual mother. A line is plotted when the mother is out at sea and a gap indicates time spent ashore. Mothers whose pups died are marked in red.

```
attendance <- transform(attendance, date.trip.left=as.Date(date.trip.left, format =
  "%d/%m/%Y"), date.trip.returned=as.Date(date.trip.returned, format = "%d/%m/%Y"))

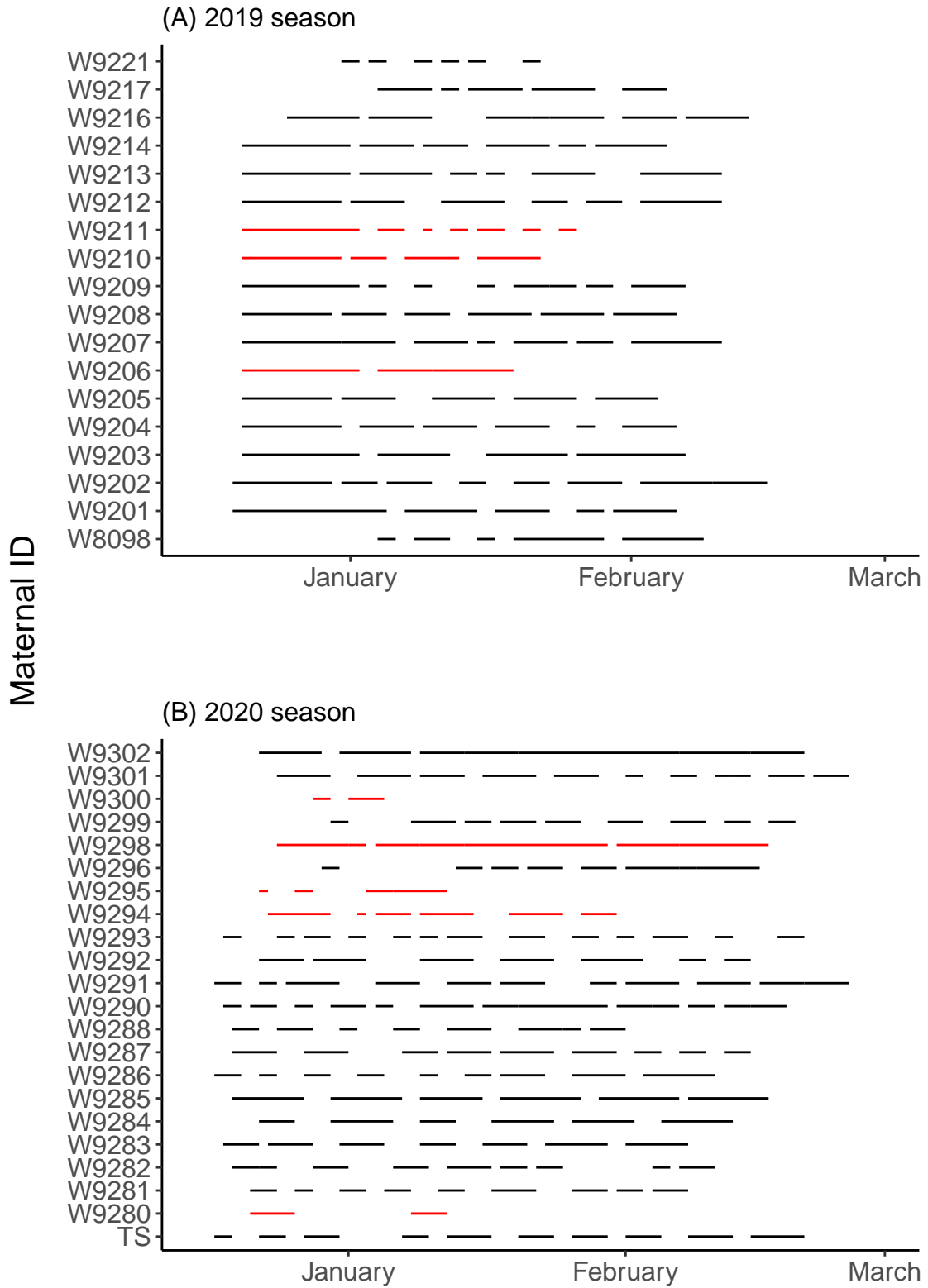
attendance <- within(attendance, seal.id<-data.frame(do.call('rbind',
  strsplit(as.character(attendance$seal.id), '/', fixed=TRUE))))

annotate_figure( ggarrange( ggplot(transform(attendance[attendance$season=="2018-2019",],
  y=order(seal.id$X2, date.trip.left)), aes(x=date.trip.left, xend=date.trip.returned,
  y=seal.id$X2, yend=seal.id$X2, col=PupDeath)) + geom_segment(na.rm=TRUE) +
  scale_color_manual(values=c("black", "red")) + scale_y_discrete() +
  scale_x_date(date_labels = "%B", date_breaks = "1 month", limits =
  as.Date(c("2018-12-15", "2019-03-01"))) + theme_classic() + theme(legend.position =
  "none", text = element_text(size=15), plot.subtitle=element_text(size = 12)) + labs(x
  = "", y = "", title = "", subtitle = "(A) 2019 season"),

ggplot(transform(attendance[attendance$season=="2019-2020",], y=order(seal.id$X2,
  date.trip.left)), aes(x=date.trip.left, xend=date.trip.returned, y=seal.id$X2,
  yend=seal.id$X2, color=PupDeath)) + geom_segment(na.rm=TRUE) +
  scale_color_manual(values=c("black", "red")) + scale_y_discrete() +
  scale_x_date(date_labels = "%B", date_breaks = "1 month", limits =
  as.Date(c("2019-12-15", "2020-03-01"))) + theme_classic() + theme(legend.position =
  "none", text = element_text(size=15), plot.subtitle=element_text(size = 12)) + labs(x
  = "", y = "", title = "", subtitle = "(B) 2020 season"),

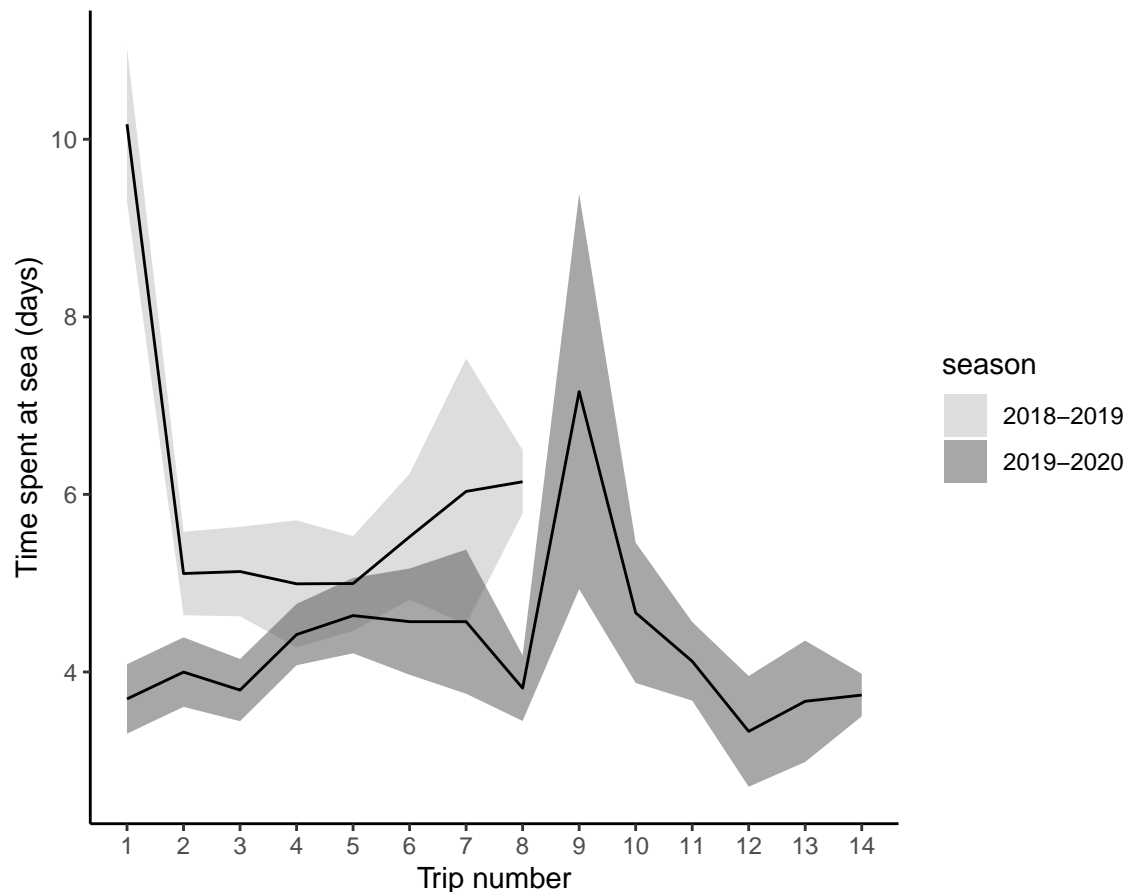
ncol = 1, nrow = 2), left = text_grob("Maternal ID", rot=90, vjust = 1.5, size = 15), top
  = text_grob("Maternal time spent at sea", size = 15))
```

Maternal time spent at sea



```
# Average time spent at sea during each foraging trip by season
ggplot(attendance, aes(y = time.at.sea.days, x = trip.number, by = season, fill =
  season)) + stat_summary(geom = "ribbon", alpha = 0.5, show.legend = TRUE) +
  stat_summary(geom = "line", show.legend = FALSE) + scale_fill_manual(values =
  c("#bcbcbc", "#515151")) + theme(text = element_text(size = 15)) + theme_classic() +
  labs(x = "Trip number", y = "Time spent at sea (days)", title = "") +
  scale_x_continuous(breaks = 1:max(attendance$trip.number))
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



Pup birth weight & no. of breeders

At SSB, twice a day the breeding colony was scanned for newborn pups, which were subsequently sexed and weighed. Mean pup birth weight per season was calculated from female pups to avoid any sex bias.

```
# 2018-19
bmass19 <- data.frame(Weight = c(4.1, 4.5, 4.5, 4.1, 4.5, 4.3, 5.5, 4.6, 3.9, 4.5, 4.8,
  4.2, 4.6, 3.3, 3.8, 3.8, 4.2, 4.4, 4, 3.4, 4.5, 4.5, 4.2, 4.4, 3.7, 4.2, 3.5, 4.7,
  3.7, 4.9, 4.6, 4, 4.1, 4.6, 4.4, 4.2, 4.6, 4.5, 5.2, 4, 4.6, 3.7, 4, 4.5, 3.6, 4.5,
```

```

    4.5, 3.9, 4.3, 4, 4.3, 4.2, 4.6, 5.4), ID = c(1:54))

bmass19_av <- groupwiseMean(Weight ~ 1, data = bmass19, conf = 0.95, digits = 3)

# 2019-20
bmass20 <- data.frame(Weight = c(4.2, 4.6, 4.9, 3.7, 5.9, 4.8, 4, 5.2, 4.6, 4.8, 4.3,
    4.6, 3.9, 5.1, 5.3, 5.4, 4.5, 5.2, 4, 4.6, 4.5, 4.8, 3.8, 4.5, 4.6, 4.1, 4.8, 4.7, 5,
    4.8, 4.9, 4.8, 5.3, 5.2, 4.6, 4, 4.9, 5, 4.3, 5, 4.5, 4.9, 4.6, 5.2, 4.8, 5.5, 5.4,
    4.6, 4.8, 4, 5.3, 4.9, 4.8, 4.6, 5, 4.6, 4.2, 4.7, 4.6, 5.4, 4.5, 3.9, 5.4, 4.8, 4.2,
    5.3, 5.1, 4.9, 5.4, 3.9), ID = c(1:70))

bmass20_av <- groupwiseMean(Weight ~ 1, data = bmass20, conf = 0.95, digits = 3)

shapiro.test(bmass19$Weight)

##
## Shapiro-Wilk normality test
##
## data:  bmass19$Weight
## W = 0.96698, p-value = 0.1416

shapiro.test(bmass20$Weight)

##
## Shapiro-Wilk normality test
##
## data:  bmass20$Weight
## W = 0.97456, p-value = 0.1647

# x has a smaller mean than y?
t.test(x = bmass19$Weight, y = bmass20$Weight, alternative = "less", var.equal = FALSE)

##
## Welch Two Sample t-test
##
## data:  bmass19$Weight and bmass20$Weight
## t = -5.2786, df = 116.56, p-value = 3.052e-07
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.3015774
## sample estimates:
## mean of x mean of y
##  4.288889  4.728571

# Percentage difference in birth weight = [ (Starting Value - Final Value) /
# |Starting Value| ] * 100
((bmass19_av$Mean - bmass20_av$Mean)/bmass19_av$Mean) * 100

## [1] -10.25641

```


The total number of pups born in the colony is equivalent to the total number of breeders in the colony; confidence interval estimates were determined based on the observed number of females mated.

```
SSB <- data.frame(Season = c("1819", "1920"), Year = c("2018-2019", "2019-2020"),
  Productivity = c(282, 409), ProductivityCI_up = c(302, 425), ProductivityCI_down =
  c(262, 393), knownF = c(132, 170), knownFbreeding = c(85, 142), proportion_breeders =
  c(0.644, 0.835), productivity_SE = c(10.02, 8.21), Weight = c(bmass19_av$Mean,
  bmass20_av$Mean), WeightCI_up = c(bmass19_av$Trad.upper, bmass20_av$Trad.upper),
  WeightCI_down = c(bmass19_av$Trad.lower, bmass20_av$Trad.lower))

# Percentage difference in the number of breeders = [ (Starting Value - Final
# Value) / |Starting Value| ] * 100
((282 - 409)/282) * 100
```

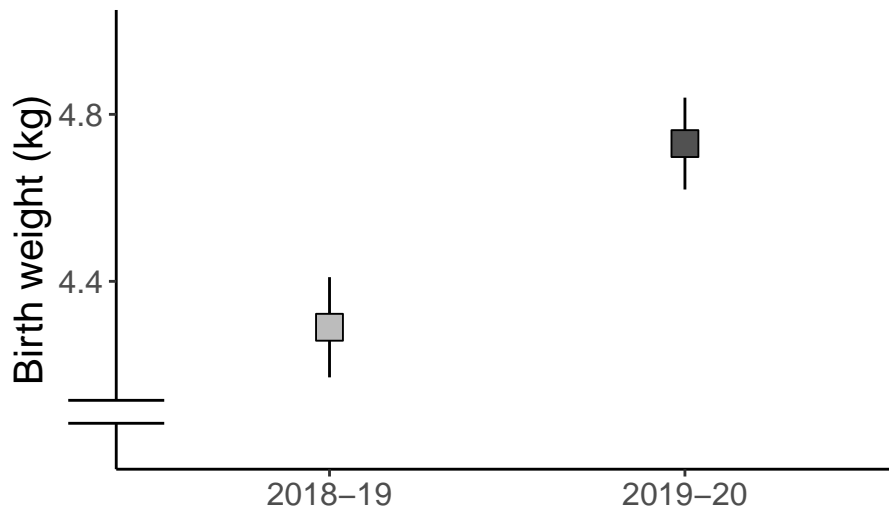
```
## [1] -45.03546
```

Plotting the total number of breeders and the average female pup birth weight (kg)

```
weight <- ggplot(data = SSB, aes(x = Year, y = Weight, group = 1, fill = Year)) +
  geom_pointrange(aes(ymin = WeightCI_down, ymax = WeightCI_up)) + # geom_line(linetype
  = 'dashed') + geom_point(shape = 22, size = 5) + scale_fill_manual(values =
  c("#bcbcbc", "#515151")) + scale_x_discrete(labels = c(`2018-2019` = "2018-19",
  `2019-2020` = "2019-20")) + theme_classic() + scale_y_continuous(limits = c(4, 5),
  breaks = c(4.4, 4.8)) + labs(y = "Birth weight (kg)", x = "", title = "(C) Pup birth
  weight", subtitle = "") + theme(text = element_text(size = 15), legend.position =
  "none", plot.margin = unit(c(10, 5, 5, 5), "mm"), plot.title = element_text(margin =
  margin(b = 0), size = 15), plot.subtitle = element_text(hjust = 0.28, margin =
  margin(t = 5, b = 10), size = 12))

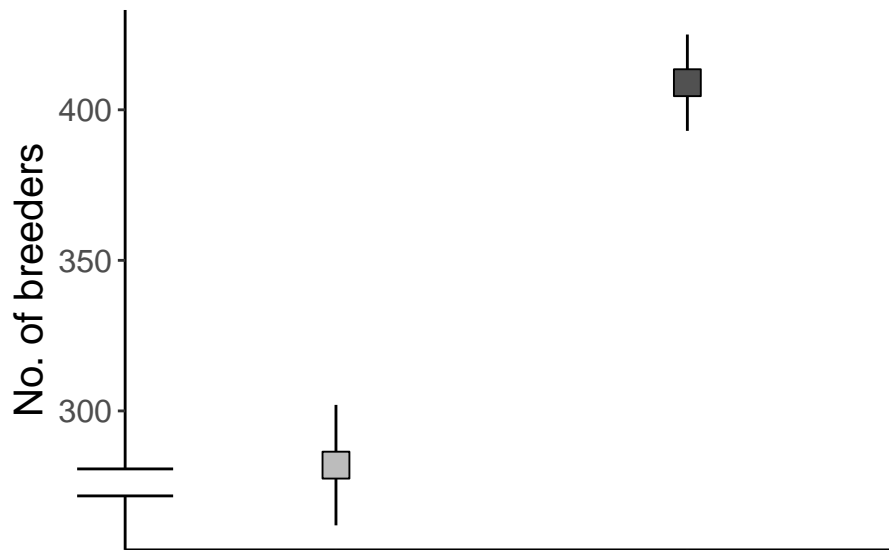
# add break on x-axis
weight <- ggplotGrob(weight)
is_yaxis <- which(weight$layout$name == "axis-l")
yaxis <- weight$grobs[[is_yaxis]]
yline <- yaxis$children[[1]]
yline$x <- unit(rep(1, 4), "npc")
yline$y <- unit(c(0, 0.1, 1, 0.15), "npc")
yline$id <- c(1, 1, 2, 2)
yline$arrow <- arrow(angle = 90)
yaxis$children[[1]] <- yline
weight$grobs[[is_yaxis]] <- yaxis
grid.newpage()
grid.draw(weight)
```

(C) Pup birth weight



```
productivity <- ggplot(data = SSB, aes(x = Year, y = Productivity, group = 1, fill =
  Year)) + geom_pointrange(aes(ymin = ProductivityCI_down, ymax = ProductivityCI_up)) +
  # geom_line(linetype = 'dashed') + geom_point(shape = 22, size = 5) +
  scale_fill_manual(values = c("#bcbcbc", "#515151")) + scale_x_discrete(labels =
  c(`2018-2019` = "2018-19", `2019-2020` = "2019-20")) + theme_classic() + labs(y =
  "No. of breeders", x = "", title = "(B) Breeding females", subtitle = "") +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), text =
  element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5, 5),
  "mm"), plot.title = element_text(margin = margin(b = 0), size = 15), plot.subtitle =
  element_text(hjust = 0.28, margin = margin(t = 5, b = 10), size = 12))
# add axis break
productivity <- ggplotGrob(productivity)
is_yaxis <- which(productivity$layout$name == "axis-l1")
yaxis <- productivity$grobs[[is_yaxis]]
yline <- yaxis$children[[1]]
yline$x <- unit(rep(1, 4), "npc")
yline$y <- unit(c(0, 0.1, 1, 0.15), "npc")
yline$id <- c(1, 1, 2, 2)
yline$arrow <- arrow(angle = 90)
yaxis$children[[1]] <- yline
productivity$grobs[[is_yaxis]] <- yaxis
grid.newpage()
grid.draw(productivity)
```

(B) Breeding females



Colony differences

Pup mortality

At both colonies, the survival of focal pups was tracked from birth until weaning. In the absence of direct proof (e.g. a corpse), mortality was assumed if the individual was not sighted for ten consecutive days.

```
# Mortality at FWB, by sex  
kable(table(pup[pup$Day == 0 & pup$Beach == "FWB" & pup$Death == "Y", ]$Sex))
```

Var1	Freq
F	7
M	9

```
# Mortality at SSB, by sex  
kable(table(pup[pup$Day == 0 & pup$Beach == "SSB" & pup$Death == "Y", ]$Sex))
```

Var1	Freq
F	5
M	1

To test if pup mortality differed significantly between the two colonies, we used a two-proportions z-test with Yates' continuity correction. Here, we test if the observed proportion of deaths at FWB is greater than the observed proportion of deaths at SSB.

```
# 16/50 die at FWB; 6/50 die at SSB
prop.test(x = c(16, 6), n = c(50, 50), correct = TRUE, alternative = "greater")
```

```
##
## 2-sample test for equality of proportions with continuity correction
##
## data: c(16, 6) out of c(50, 50)
## X-squared = 4.7203, df = 1, p-value = 0.0149
## alternative hypothesis: greater
## 95 percent confidence interval:
## 0.04775541 1.00000000
## sample estimates:
## prop 1 prop 2
## 0.32 0.12
```

To test if breeding colony and focal pup birth weight were significant predictors of pup mortality, we used a generalized linear model (GLM) with a binomial distribution.

```
pup$Death <- ifelse(pup$Death == "Y", 1, 0)

m.death <- glm(Death ~ Weight + Beach + Sex + CI_MumAvg, data = pup[pup$Day == 0, ],
  family = binomial())

m.death1 <- glm(Death ~ Weight + Beach + CI_MumAvg, data = pup[pup$Day == 0, ], family =
  binomial())

m.death2 <- glm(Death ~ Weight + Beach, data = pup[pup$Day == 0, ], family = binomial())

m.death3 <- glm(Death ~ Weight, data = pup[pup$Day == 0, ], family = binomial())

AIC(m.death, m.death1, m.death2, m.death3)
```

```
##          df          AIC
## m.death   5 102.43550
## m.death1  4 100.92528
## m.death2  3  99.61527
## m.death3  2 102.77128
```

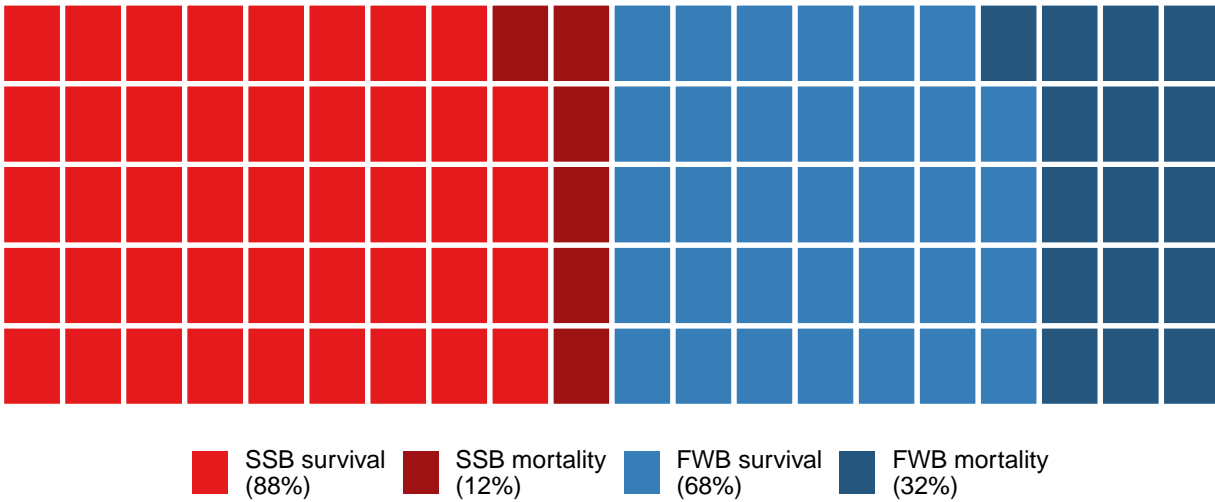
```
summary(m.death2)
```

```
##
## Call:
## glm(formula = Death ~ Weight + Beach, family = binomial(), data = pup[pup$Day ==
## 0, ])
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1764  -0.7716  -0.4934  -0.3161   2.2892
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) 3.1391 1.7563 1.787 0.0739 .
## Weight -0.7140 0.3212 -2.223 0.0262 *
## BeachSSB -1.1857 0.5447 -2.177 0.0295 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 104.882 on 98 degrees of freedom
## Residual deviance: 93.615 on 96 degrees of freedom
## (1 observation deleted due to missingness)
## AIC: 99.615
##
## Number of Fisher Scoring iterations: 4
```

```
survive <- c(`SSB survival\n(88%)` = 44, `SSB mortality\n(12%)` = 6, `FWB
survival\n(68%)` = 34, `FWB mortality\n(32%)` = 16)

waffle(survive, rows = 5, size = 1, legend_pos = "bottom", colors = c("#e41a1c",
"#9f1214", "#377eb8", "#25567d"), equal = FALSE) + theme(legend.text =
element_text(size = 10))
```



Local density

At every capture every 10 days, the local density of each pup, determined as the total number of individuals within two meters' radius, was noted. Here, we plot the density distribution to visually assess differences between FWB and SSB.

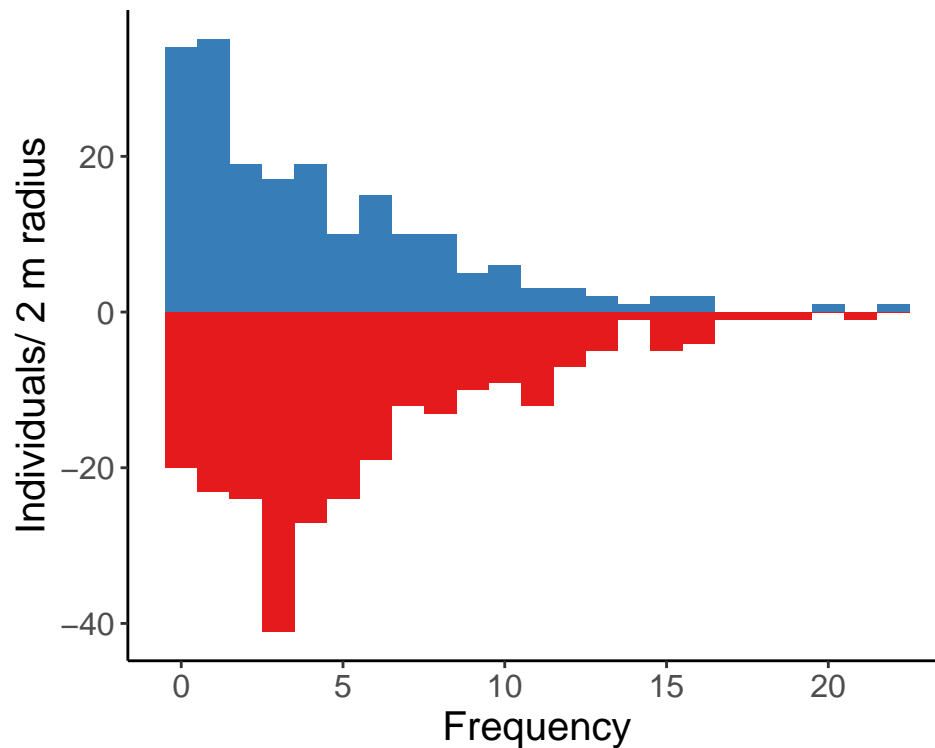
```
# what is the range of local density at FWB?
sort(table(pup$LocalDensity[pup$Beach == "FWB"]), decreasing = TRUE)
```

```
##
## 1 0 2 4 3 6 5 7 8 10 9 11 12 13 15 16 14 20 22
## 35 34 19 19 17 15 10 10 10 6 5 3 3 2 2 2 1 1 1
```

```
# what is the range of local density at SSB?
sort(table(pup$LocalDensity[pup$Beach == "SSB"]), decreasing = TRUE)
```

```
##
## 3 4 2 5 1 0 6 8 7 11 9 10 12 13 15 16 14 17 18 19 21
## 41 27 24 24 23 20 19 13 12 12 10 9 7 5 5 4 1 1 1 1 1
```

```
ggplot() + geom_histogram(aes(x = pup$LocalDensity[pup$Beach == "FWB"], y = stat(count)),
  fill = "#377eb8", binwidth = 1, na.rm = TRUE) + # geom_label(aes(x=20, y=0.05,
  label='FWB'), color='#377eb8') + geom_histogram(aes(x = pup$LocalDensity[pup$Beach ==
  "SSB"], y = -stat(count)), fill = "#e41a1c", binwidth = 1, na.rm = TRUE) + #
  geom_label(aes(x=20, y=-0.05, label='SSB'), color='#e41a1c') + theme_classic() +
  xlab("Frequency") + ylab("Individuals/ 2 m radius") + theme(text = element_text(size
  = 15))
```



Pups

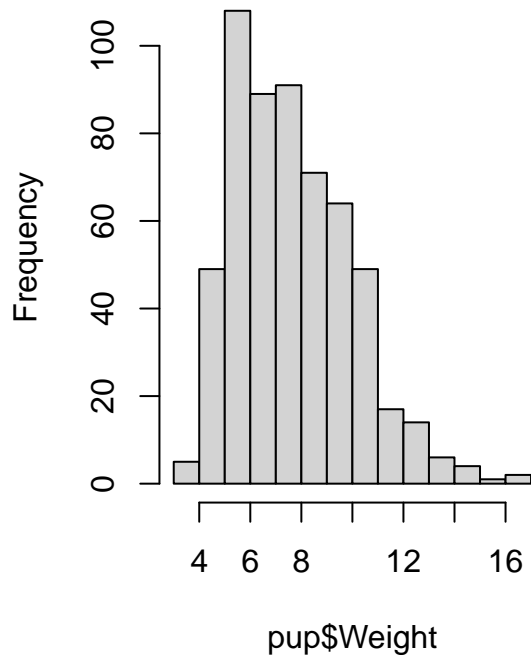
Weight

The weight of all focal pups was measured every 10 days.

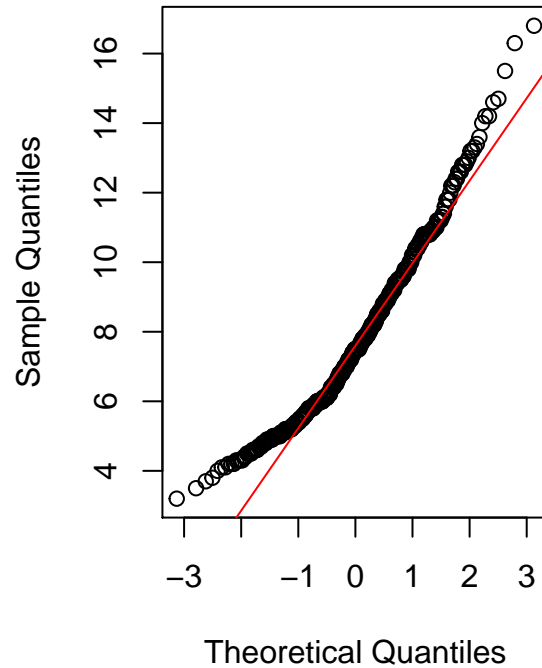
Here, we are testing if age, sex, maternal weight just after birth, season, colony of birth and/or an interaction thereof are significant predictors of the weight of pups.

```
# test normality of condition index
par(mfrow = c(1, 2))
hist(pup$Weight, main = "Histogram of pup weight")
qqnorm(pup$Weight)
qqline(pup$Weight, col = "red")
```

Histogram of pup weight



Normal Q-Q Plot



```
shapiro.test(pup$Weight)
```

```
##
## Shapiro-Wilk normality test
##
## data: pup$Weight
## W = 0.96026, p-value = 2.78e-11
```

Next, we built a linear mixed model fitted by restricted maximum-likelihood (REML). `lmerTest` implements a backward elimination process to simplify the model; the statistical significance of fixed predictors is assessed using Type III ANOVA (Satterthwaite's approximation of denominator degrees of freedom).

```
w.full <- lmer(Weight ~ Day_Actual + Beach * Season + Weight_MumBirth + Sex + (1 |
  ID_Pup), data = pup)
w.step <- step(w.full)
w.step
```

```

## Backward reduced random-effect table:
##
##           Eliminated npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                9 -871.95 1761.9
## (1 | ID_Pup)          0   8 -992.17 2000.3 240.44 1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Backward reduced fixed-effect table:
## Degrees of freedom method: Satterthwaite
##
##           Eliminated  Sum Sq Mean Sq NumDF  DenDF  F value  Pr(>F)
## Beach:Season          1    1.32   1.32    1  94.71   1.5368 0.218152
## Season                 2    0.02   0.02    1  96.08   0.0250 0.874740
## Beach                  3    1.45   1.45    1  97.11   1.6826 0.197646
## Day_Actual            0 1502.76 1502.76    1 487.75 1747.4436 < 2.2e-16 ***
## Weight_MumBirth       0    9.52   9.52    1  98.09  11.0740 0.001234 **
## Sex                   0   14.21  14.21    1 115.96  16.5264 8.772e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model found:
## Weight ~ Day_Actual + Weight_MumBirth + Sex + (1 | ID_Pup)

```

```

w.best <- get_model(w.step)
summary(w.best)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Weight ~ Day_Actual + Weight_MumBirth + Sex + (1 | ID_Pup)
## Data: pup
##
## REML criterion at convergence: 1744.9
##
## Scaled residuals:
##   Min      1Q  Median      3Q      Max
## -3.3357 -0.6224  0.0000  0.6349  3.8014
##
## Random effects:
## Groups Name Variance Std.Dev.
## ID_Pup (Intercept) 1.00 1.0001
## Residual 0.86 0.9274
## Number of obs: 570, groups: ID_Pup, 100
##
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 2.243e+00 8.365e-01 9.858e+01 2.681 0.00860 **
## Day_Actual 7.980e-02 1.909e-03 4.878e+02 41.802 < 2e-16 ***
## Weight_MumBirth 7.879e-02 2.368e-02 9.809e+01 3.328 0.00123 **
## SexM 8.570e-01 2.108e-01 1.160e+02 4.065 8.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:

```

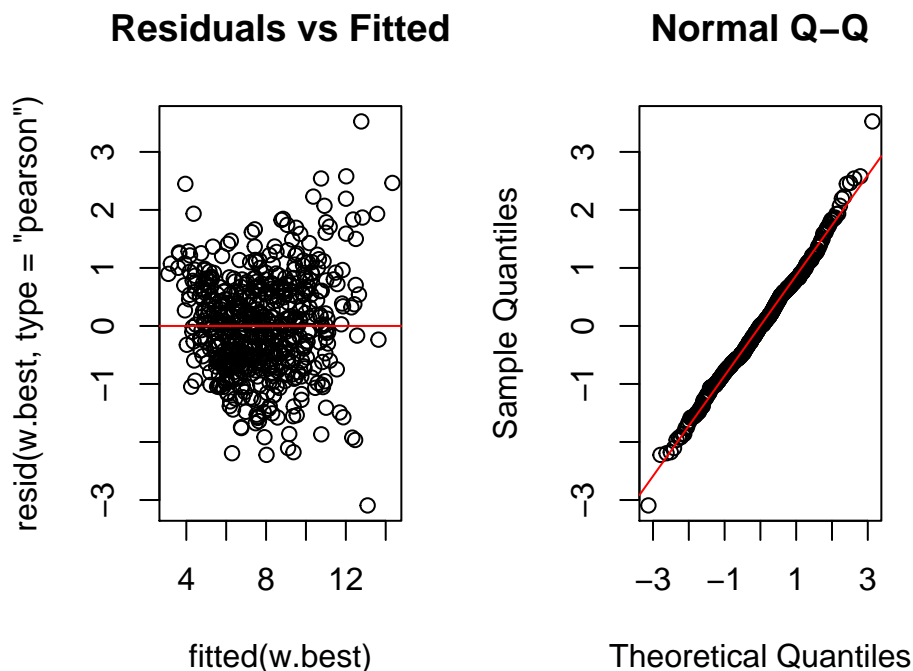


```
##          (Intr) Dy_Act Wgh_MB
## Day_Actual -0.021
## Wght_MmBrth -0.982 -0.036
## SexM      -0.012 -0.011 -0.116
```

Check residuals

```
# Check for linearity and equal error variances. Points should be randomly
# scattered around zero for the entire range of fitted values
par(mfrow = c(1, 2))
plot(fitted(w.best), resid(w.best, type = "pearson"), main = "Residuals vs Fitted")
abline(0, 0, col = "red")

# Check if residuals are normally distributed. Points should follow the straight
# line.
qqnorm(resid(w.best), main = "Normal Q-Q")
qqline(resid(w.best), col = "red")
```



```
# Additional check for homogeneity of variance - Levene's Test
pup.rmna <- pup[complete.cases(pup[, c("Weight", "Day_Actual", "Weight_MumBirth", "Sex",
  "ID_Pup")]), ] # build new dataset without NAs

pup.rmna$w.best.Res <- residuals(w.best) # extracts the residuals and adds them to the
  data table
pup.rmna$Abs.w.best.Res <- abs(pup.rmna$w.best.Res) # absolute value of the residuals
pup.rmna$w.best.Res2 <- pup.rmna$Abs.w.best.Res^2 # square the absolute values of the
  residuals to provide a more robust estimate
Levene.Model.F <- lm(w.best.Res2 ~ ID_Pup, data = pup.rmna) # ANOVA of the squared
```

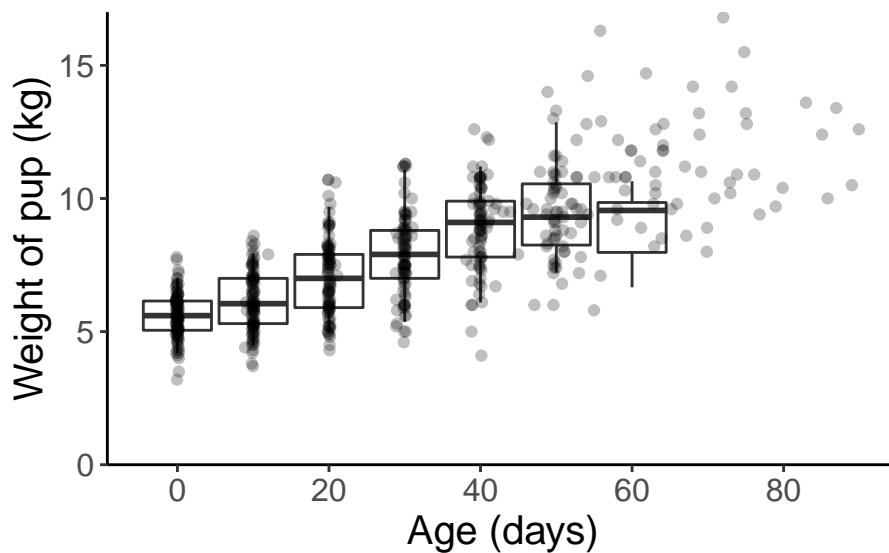
```
residuals
anova(Levene.Model.F)
```

```
## Analysis of Variance Table
##
## Response: w.best.Res2
##           Df Sum Sq Mean Sq F value    Pr(>F)
## ID_Pup     99 181.89  1.8373  1.4153 0.009752 **
## Residuals 470 610.12  1.2981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot() + stat_summary(fun.data = quantiles_95, geom="boxplot", data =
  pup[which(pup$Day_Actual<=51),], aes(y = Weight, x = Day, group = Day), fill =
  "#ffffff", na.rm=TRUE) + geom_point(data = pup, aes(y = Weight, x = Day_Actual, group
  = Day, fill = Beach), position = position_jitterdodge(), na.rm=TRUE, alpha = 0.25) +
  theme_classic() + theme(text = element_text(size=15), legend.position = "none",
  plot.margin=unit(c(10,5,5,5),"mm"), plot.title=element_text(margin=margin(b=0), size
  = 15), plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size =
  12)) + labs(x = "Age (days)", y = "Weight of pup (kg)", title = "(A) Age of pup",
  subtitle = expression(paste("estimate = 0.080, ", italic("p"), " < 0.001"))) +
  scale_x_continuous(minor_breaks=NULL, breaks = c(0,20,40,60,80),
  labels=c(0,20,40,60,80)) + scale_y_continuous(expand = c(0, 0), limits = c(0, 17),
  breaks = c(0, 5, 10, 15))
```

(A) Age of pup

estimate = 0.080, $p < 0.001$



```
ggplot(data = pup) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y =
  Weight, x = Sex, fill = Sex), na.rm=TRUE, fill = c("#ffffff", "#ffffff"), #color =
```

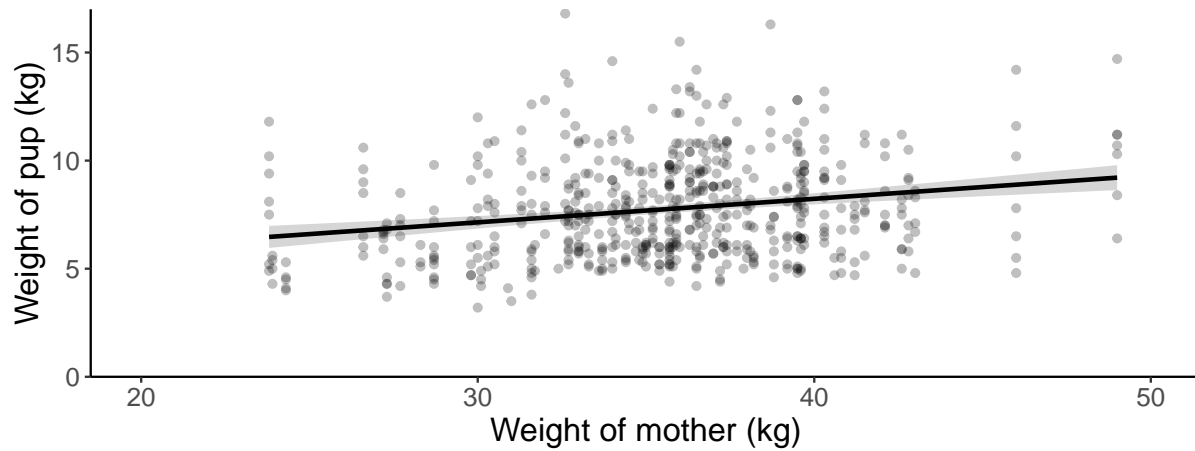
```
c("#bcbcbc", "#515151"), alpha = 0.25, width = 0.35) + geom_point(aes(y = Weight, x = Sex, fill = Sex), na.rm=TRUE, position = position_jitterdodge(), alpha = 0.25) + theme_classic() + theme(text = element_text(size=15), legend.position = "none", plot.margin=unit(c(10,5,5,5),"mm"), plot.title=element_text(margin=b=0), size = 15), plot.subtitle=element_text(hjust = 0.28, margin=margint=5, b = 10), size = 12)) + labs(x = "Sex", y = "Weight of pup (kg)", title = "(B) Sex of pup", subtitle = expression(paste("estimate = 0.857, ", italic("p"), " < 0.001"))) + scale_x_discrete(breaks = c("F", "M"), labels=c("Female","Male")) + scale_y_continuous(expand = c(0, 0), limits = c(0, 17), breaks = c(0, 5, 10, 15))
```



```
ggplot(data = pup) + geom_smooth(aes(y = Weight, x = Weight_MumBirth), method = "lm", formula = y ~ x, color = "black", na.rm = TRUE) + geom_point(aes(y = Weight, x = Weight_MumBirth), na.rm = TRUE, alpha = 0.25) + theme_classic() + theme(text = element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5, 5), "mm"), plot.title = element_text(margin = margin(b = 0.1), size = 15), plot.subtitle = element_text(hjust = 0.06, margin = margin(t = 5, b = 10), size = 12)) + labs(x = "Weight of mother (kg)", y = "Weight of pup (kg)", title = "(C) Postpartum weight of mother", subtitle = expression(paste("estimate = 0.079, ", italic("p"), " = 0.001"))) + scale_x_continuous(limits = c(20, 50), breaks = c(20, 30, 40, 50)) + scale_y_continuous(expand = c(0, 0), limits = c(0, 17), breaks = c(0, 5, 10, 15))
```

(C) Postpartum weight of mother

estimate = 0.079, $p = 0.001$



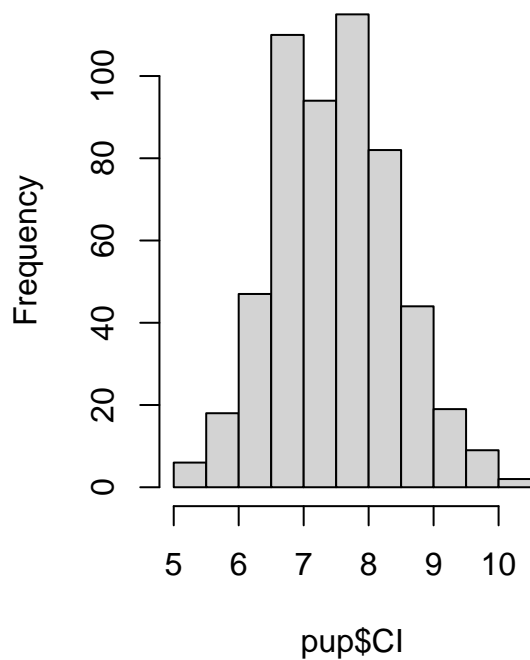
Condition index

The condition index (CI) of all focal individuals was calculated using a scaled mass index according to (Peig and Green, 2009). This method uses log-transformed scaling between body length and mass in relation to the average body length for the population, thereby taking individual differences in relative size into consideration.

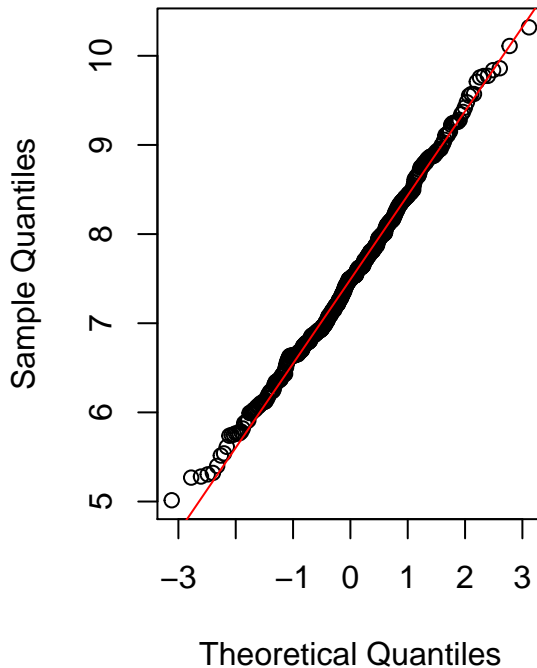
We are testing if age, sex, maternal CI just after birth, season, colony of birth and/or an interaction thereof are significant predictors of the CI of pups.

```
par(mfrow = c(1, 2))
hist(pup$CI, main = "Histogram of pup CI")
qqnorm(pup$CI)
qqline(pup$CI, col = "red")
```

Histogram of pup CI



Normal Q-Q Plot



```
shapiro.test(pup$CI) # test normality
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: pup$CI  
## W = 0.99548, p-value = 0.1148
```

For this, we built a linear mixed model fitted by restricted maximum-likelihood (REML). `lmerTest` implements a backward elimination process to simplify the model; the statistical significance of fixed predictors is assessed using Type III ANOVA (Satterthwaite's approximation of denominator degrees of freedom).

```
m.full <- lmer(CI ~ Day_Actual * Beach * Season + CI_MumAvg + Sex + (1 | ID_Pup), data =  
  pup)  
m.step <- step(m.full)  
m.best <- get_model(m.step)  
  
summary(m.best)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: CI ~ Day_Actual + Beach + Season + (1 | ID_Pup) + Day_Actual:Beach +  
## Beach:Season  
## Data: pup
```

```

##
## REML criterion at convergence: 1449.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.09512 -0.63901 -0.04608  0.65151  3.09055
##
## Random effects:
##  Groups   Name      Variance Std.Dev.
##  ID_Pup   (Intercept) 0.1058  0.3253
##  Residual                0.7150  0.8456
## Number of obs: 546, groups: ID_Pup, 100
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      6.975165   0.117529 178.254133  59.348 < 2e-16 ***
## Day_Actual        0.010378   0.002776 490.749001   3.739 0.000207 ***
## BeachSSB          0.535203   0.163013 171.452844   3.283 0.001244 **
## Season1920        0.345467   0.148083 108.320832   2.333 0.021499 *
## Day_Actual:BeachSSB -0.007532   0.003650 483.602475  -2.064 0.039586 *
## BeachSSB:Season1920 -0.443971   0.200918  96.448228  -2.210 0.029491 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) Dy_Act BchSSB Ss1920 D_A:BS
## Day_Actual   -0.431
## BeachSSB     -0.721  0.311
## Season1920   -0.568 -0.180  0.410
## Dy_Act:BSSB  0.328 -0.761 -0.475  0.137
## BcSSB:S1920  0.419  0.133 -0.564 -0.737 -0.134

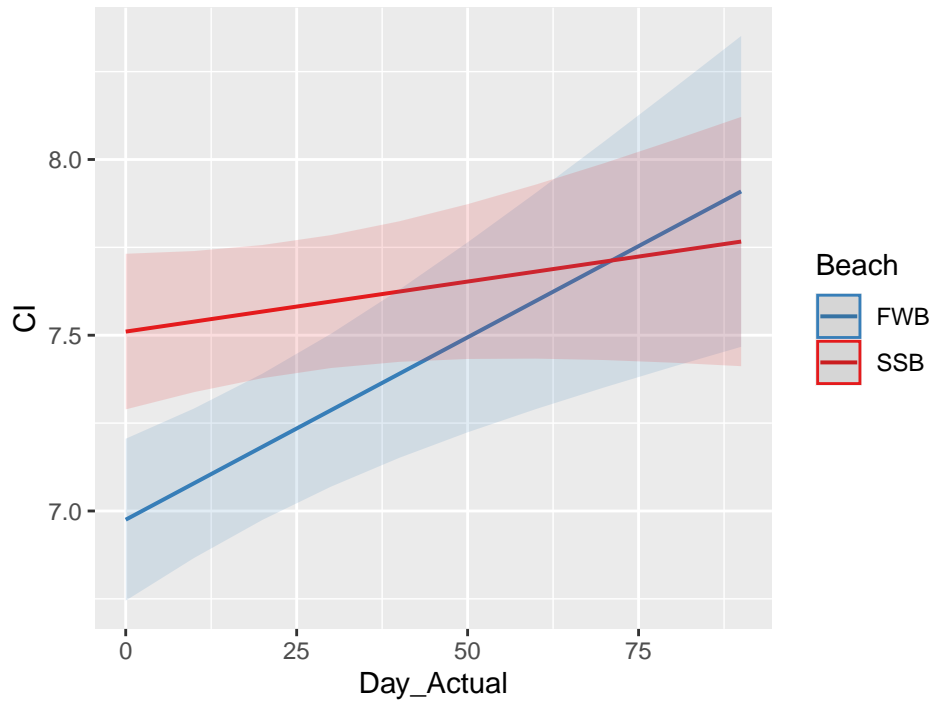
```

Look at interactions

```
plot_model(m.best, type = "int", show.data = FALSE, colors = c("#377eb8", "#e41a1c"))
```

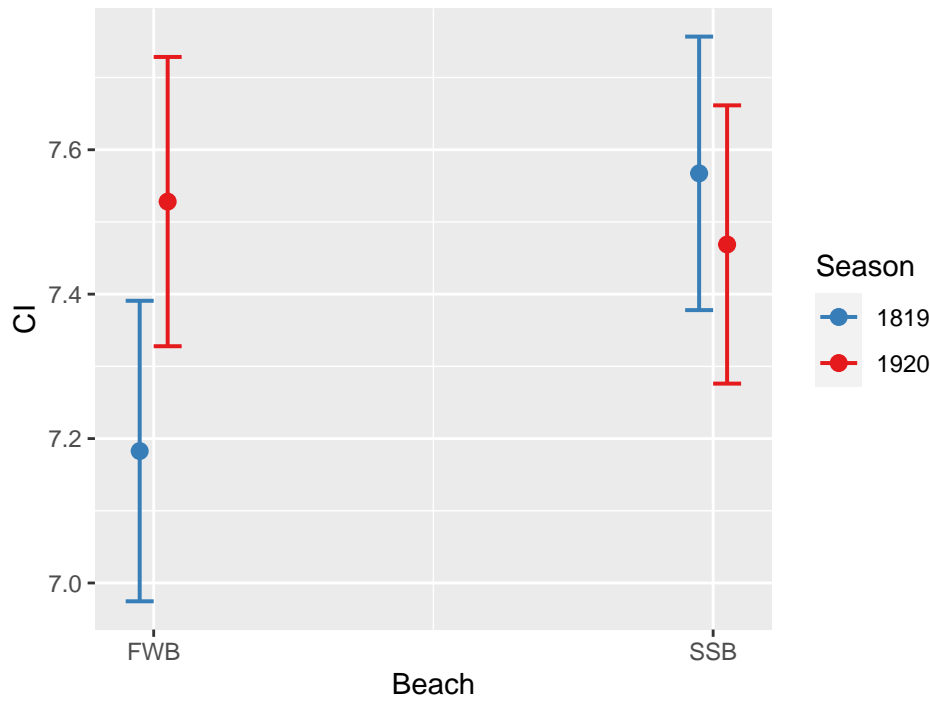
```
## [[1]]
```

Predicted values of CI



```
##  
## [[2]]
```

Predicted values of CI



```
emmeans(m.best, pairwise ~ Beach * Season, adjust = "tukey") #post-hoc
```

```
## $emmeans
## Beach Season emmean SE df lower.CL upper.CL
## FWB 1819 7.25 0.1085 125.2 6.97 7.52
## SSB 1819 7.59 0.0959 87.5 7.34 7.83
## FWB 1920 7.59 0.1000 85.6 7.34 7.85
## SSB 1920 7.49 0.0960 80.9 7.24 7.73
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 4 estimates
##
## $contrasts
## contrast estimate SE df t.ratio p.value
## FWB 1819 - SSB 1819 -0.33642 0.145 106.3 -2.323 0.0992
## FWB 1819 - FWB 1920 -0.34547 0.148 108.2 -2.331 0.0974
## FWB 1819 - SSB 1920 -0.23792 0.145 102.4 -1.642 0.3599
## SSB 1819 - FWB 1920 -0.00905 0.139 86.5 -0.065 0.9999
## SSB 1819 - SSB 1920 0.09850 0.136 84.5 0.725 0.8868
## FWB 1920 - SSB 1920 0.10755 0.139 83.3 0.776 0.8652
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```
emmeans(m.best, pairwise ~ Season * Beach * Day_Actual, at = list(Day_Actual = 0), adjust = "tukey")
```

```
## $emmeans
## Season Beach Day_Actual emmean SE df lower.CL upper.CL
## 1819 FWB 0 6.98 0.118 178 6.68 7.27
## 1920 FWB 0 7.32 0.126 222 7.00 7.64
## 1819 SSB 0 7.51 0.113 164 7.23 7.79
## 1920 SSB 0 7.41 0.119 186 7.11 7.71
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 4 estimates
##
## $contrasts
## contrast estimate SE df t.ratio p.value
## 1819 FWB 0 - 1920 FWB 0 -0.3455 0.148 108.2 -2.331 0.0974
## 1819 FWB 0 - 1819 SSB 0 -0.5352 0.163 171.3 -3.282 0.0068
## 1819 FWB 0 - 1920 SSB 0 -0.4367 0.167 181.9 -2.615 0.0472
## 1920 FWB 0 - 1819 SSB 0 -0.1897 0.170 193.5 -1.119 0.6780
## 1920 FWB 0 - 1920 SSB 0 -0.0912 0.173 203.8 -0.526 0.9526
## 1819 SSB 0 - 1920 SSB 0 0.0985 0.136 84.5 0.725 0.8868
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

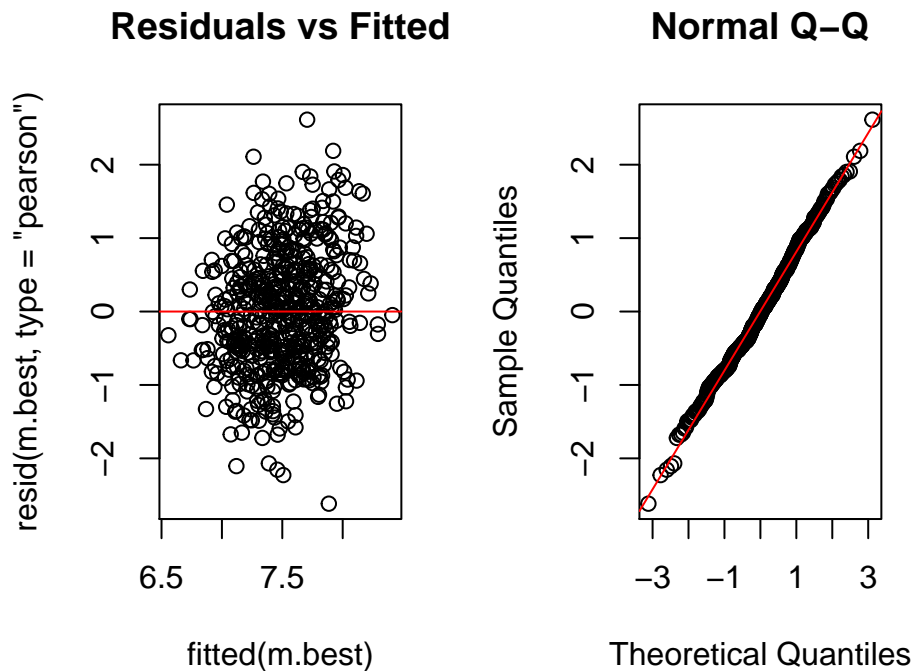
Check residuals


```

# Check for linearity and equal error variances. Points should be randomly
# scattered around zero for the entire range of fitted values
par(mfrow = c(1, 2))
plot(fitted(m.best), resid(m.best, type = "pearson"), main = "Residuals vs Fitted")
abline(0, 0, col = "red")

# Check if residuals are normally distributed. Points should follow the straight
# line.
qqnorm(resid(m.best), main = "Normal Q-Q")
qqline(resid(m.best), col = "red")

```



```

# Additional check for homogeneity of variance - Levene's Test
pup.rmna <- pup[complete.cases(pup[, c("CI", "Day_Actual", "Beach", "Season",
  "ID_Pup")]), ]

pup.rmna$m.best.Res <- residuals(m.best) # extract residuals and adds them to our data
table
pup.rmna$Abs.m.best.Res <- abs(pup.rmna$m.best.Res) # absolute value of the residuals
pup.rmna$m.best.Res2 <- pup.rmna$Abs.m.best.Res^2 # square the absolute values of the
residuals to provide the more robust estimate
Levene.Model.F <- lm(m.best.Res2 ~ ID_Pup, data = pup.rmna) # ANOVA of the squared
residuals
anova(Levene.Model.F)

```

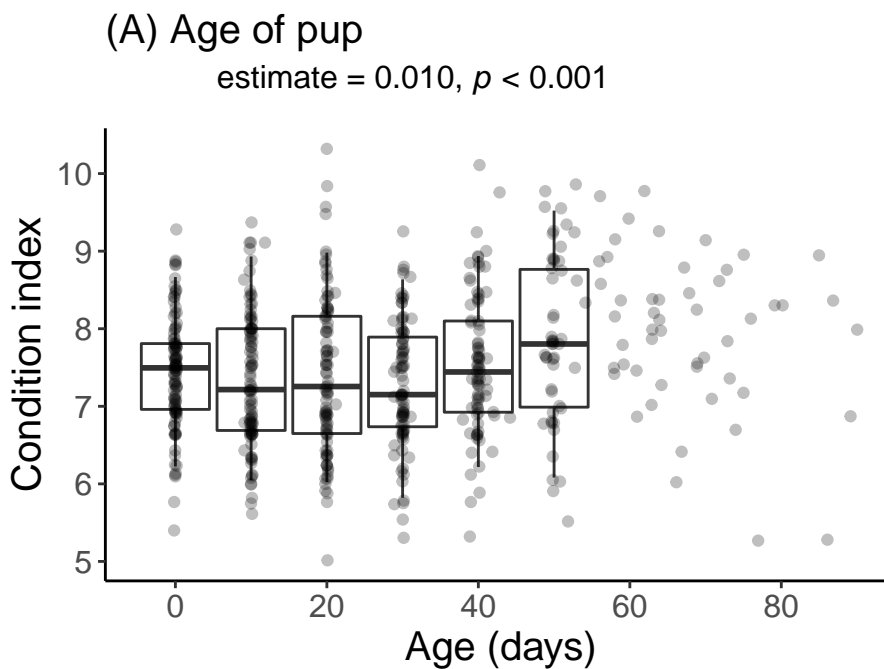
```

## Analysis of Variance Table
##
## Response: m.best.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)

```

```
## ID_Pup      99 73.36 0.74099 0.8765 0.7862
## Residuals 446 377.06 0.84544
```

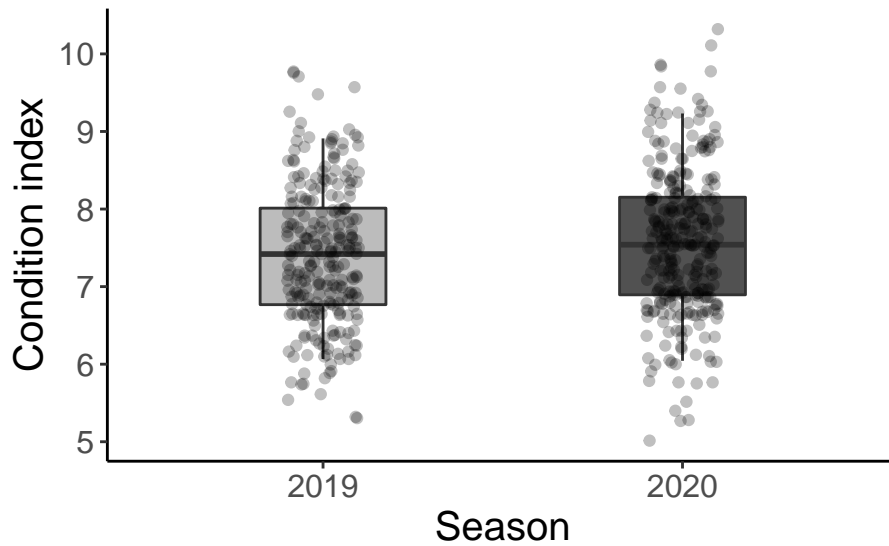
```
ggplot() + stat_summary(fun.data = quantiles_95, geom = "boxplot", data =
  pup[which(pup$Day_Actual <= 51), ], aes(y = CI, x = Day, group = Day), fill =
  "#ffffff", na.rm = TRUE) + geom_point(data = pup, aes(y = CI, x = Day_Actual, group =
  Day, fill = Beach), position = position_jitterdodge(), na.rm = TRUE, alpha = 0.25) +
  theme_classic() + theme(text = element_text(size = 15), legend.position = "none",
  plot.margin = unit(c(10, 5, 5, 5), "mm"), plot.title = element_text(margin = margin(b
  = 0), size = 15), plot.subtitle = element_text(hjust = 0.28, margin = margin(t = 5, b
  = 10), size = 12)) + labs(x = "Age (days)", y = "Condition index", title = "(A) Age
  of pup", subtitle = expression(paste("estimate = 0.010, ", italic("p"), " < 0.001")))
  + scale_x_continuous(minor_breaks = NULL, breaks = c(0, 20, 40, 60, 80), labels =
  c(0, 20, 40, 60, 80))
```



```
ggplot(data = pup) + stat_summary(fun.data = quantiles_95, geom = "boxplot", aes(y = CI,
  x = Season, fill = Season), na.rm = TRUE, fill = c("#bcbcbc", "#515151"), width =
  0.35) + geom_point(aes(y = CI, x = Season, fill = Season), na.rm = TRUE, position =
  position_jitterdodge(), alpha = 0.25) + theme_classic() + theme(text =
  element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5, 5),
  "mm"), plot.title = element_text(margin = margin(b = 0), size = 15), plot.subtitle =
  element_text(hjust = 0.28, margin = margin(t = 5, b = 10), size = 12)) + labs(x =
  "Season", y = "Condition index", title = "(B) Season of pup birth", subtitle =
  expression(paste("estimate = 0.345, ", italic("p"), " = 0.020"))) +
  scale_x_discrete(breaks = c("1819", "1920"), labels = c("2019", "2020"))
```

(B) Season of pup birth

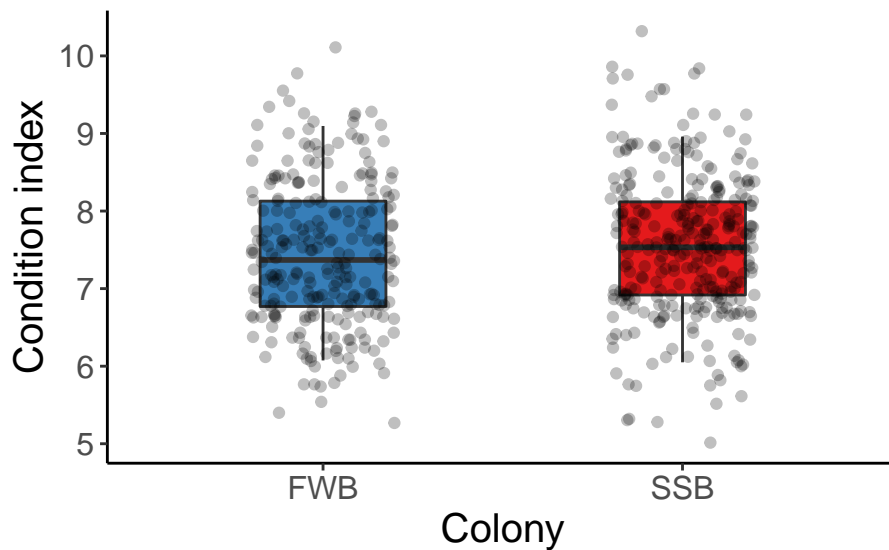
estimate = 0.345, $p = 0.020$



```
ggplot(data = pup) + stat_summary(fun.data = quantiles_95, geom = "boxplot", aes(y = CI,
  x = Beach, fill = Beach), na.rm = TRUE, fill = c("#377eb8", "#e41a1c"), width = 0.35)
+ geom_point(aes(y = CI, x = Beach, fill = Beach), na.rm = TRUE, position =
  position_jitterdodge(), alpha = 0.25) + theme_classic() + theme(text =
  element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5, 5),
  "mm"), plot.title = element_text(margin = margin(b = 0), size = 15), plot.subtitle =
  element_text(hjust = 0.28, margin = margin(t = 5, b = 10), size = 12)) + labs(x =
  "Colony", y = "Condition index", title = "(C) Social environment of pup", subtitle =
  expression(paste("estimate = 0.535, ", italic("p"), " = 0.001")))
```

(C) Social environment of pup

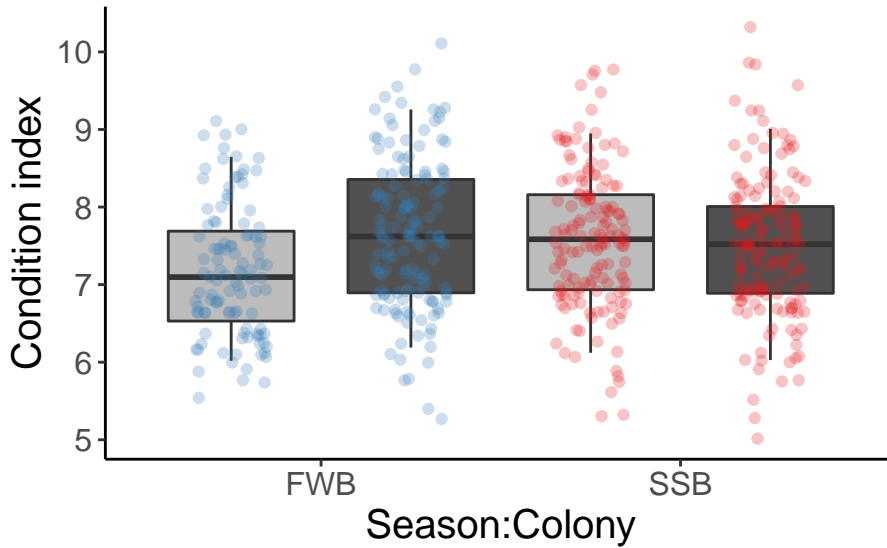
estimate = 0.535, $p = 0.001$



```
ggplot(data = pup) + stat_summary(fun.data = quantiles_95, geom = "boxplot", aes(y = CI,
  x = Beach, fill = Season), na.rm = TRUE, position = position_dodge(width = 1), width
  = 0.7) + geom_point(aes(y = CI, x = Beach, fill = Season, color = Beach), na.rm =
  TRUE, position = position_jitterdodge(dodge.width = 1), alpha = 0.25) +
  theme_classic() + theme(text = element_text(size = 15), legend.position = "none",
  plot.margin = unit(c(10, 5, 5, 5), "mm"), plot.title = element_text(margin = margin(b
  = 0), size = 15), plot.subtitle = element_text(hjust = 0.28, margin = margin(t = 5, b
  = 10), size = 12)) + labs(x = "Season:Colony", y = "Condition index", title = "(D)
  Interaction term", subtitle = expression(paste("estimate = -0.444, ", italic("p"), "
  = 0.027"))) + scale_fill_manual(values = c("#bcbcbc", "#515151"), labels = c("2019",
  "2020"), name = "") + scale_color_manual(values = c("#377eb8", "#e41a1c"))
```

(D) Interaction term

estimate = -0.444, $p = 0.027$

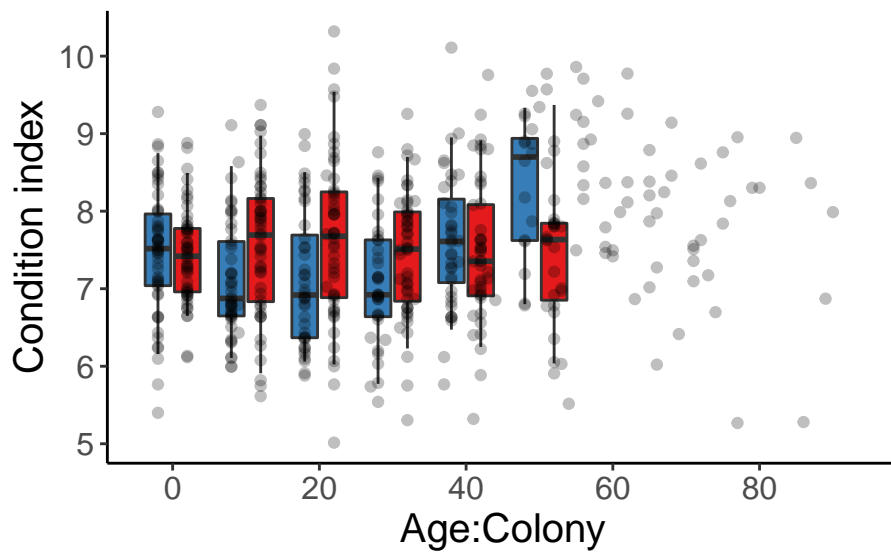


```
ggplot() + stat_summary(fun.data = quantiles_95, geom = "boxplot", data =  
  pup[which(pup$Day_Actual <= 51), ], aes(y = CI, x = Day, fill = Beach, group =  
  interaction(Day, Beach)), na.rm = TRUE, position = position_dodge(width = 8), width =  
  7) + geom_point(data = pup, aes(y = CI, x = Day_Actual, fill = Beach), position =  
  position_dodge(width = 8), na.rm = TRUE, alpha = 0.25) + theme_classic() + theme(text  
  = element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5,  
  5), "mm"), plot.title = element_text(margin = margin(b = 0.1), size = 15),  
  plot.subtitle = element_text(hjust = 0.06, margin = margin(t = 5, b = 10), size =  
  12)) + labs(x = "Age:Colony", y = "Condition index", title = "(E) Interaction term",  
  subtitle = expression(paste("estimate = -0.008, ", italic("p"), " = 0.039"))) +  
  scale_color_manual(values = c("#377eb8", "#e41a1c")) + scale_fill_manual(values =  
  c("#377eb8", "#e41a1c")) + scale_x_continuous(minor_breaks = NULL, breaks = c(0, 20,  
  40, 60, 80), labels = c(0, 20, 40, 60, 80))
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

(E) Interaction term

estimate = -0.008, $p = 0.039$



Mothers

Condition index

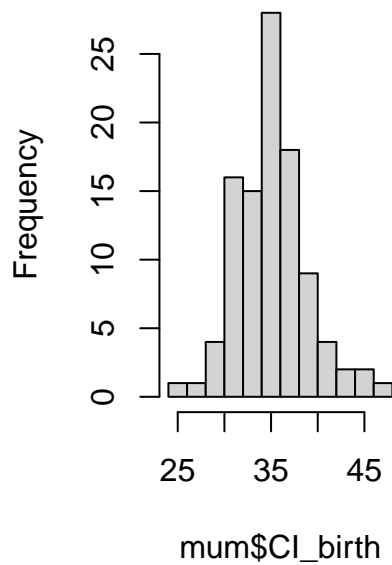
Here we are interested to test if maternal CI at birth differs between colonies and/or seasons.

```
summary(mum$CI_birth)
```

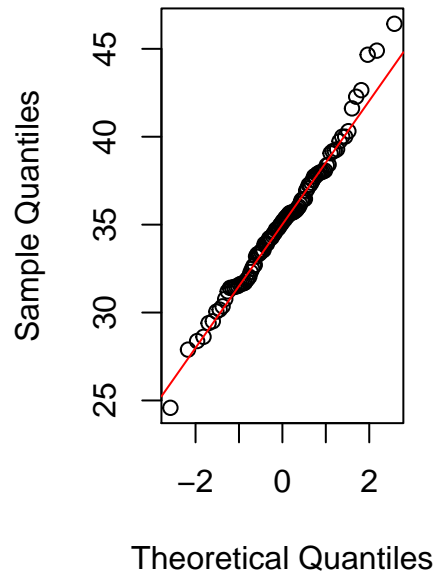
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 24.58  32.65   35.15   35.15  37.39   46.42
```

```
par(mfrow = c(1, 2))
hist(mum$CI_birth, main = "Histogram of mum CI")
qqnorm(mum$CI_birth)
qqline(mum$CI_birth, col = "red")
```

Histogram of mum CI



Normal Q-Q Plot



```
shapiro.test(mum$CI_birth)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  mum$CI_birth  
## W = 0.98416, p-value = 0.2694
```

Next, we built a linear model. Backwards elimination of fixed effects; best model selection with AIC.

```
mm.1 <- lm(CI_birth ~ Beach * Season, data = mum)  
mm.2 <- lm(CI_birth ~ Beach + Season, data = mum)  
mm.3 <- lm(CI_birth ~ Beach, data = mum)  
mm.4 <- lm(CI_birth ~ Season, data = mum)
```

```
anova(mm.1, mm.2, mm.3, mm.4)
```

```
## Analysis of Variance Table  
##  
## Model 1: CI_birth ~ Beach * Season  
## Model 2: CI_birth ~ Beach + Season  
## Model 3: CI_birth ~ Beach  
## Model 4: CI_birth ~ Season  
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)  
## 1      97 1342.3  
## 2      98 1358.8 -1   -16.444 1.1883 0.2784  
## 3      99 1374.3 -1   -15.470 1.1179 0.2930  
## 4      99 1360.8  0    13.481
```

```
AIC(mm.1, mm.2, mm.3, mm.4)
```

```
##      df      AIC
## mm.1  5 557.9186
## mm.2  4 557.1483
## mm.3  3 556.2918
## mm.4  3 555.2961
```

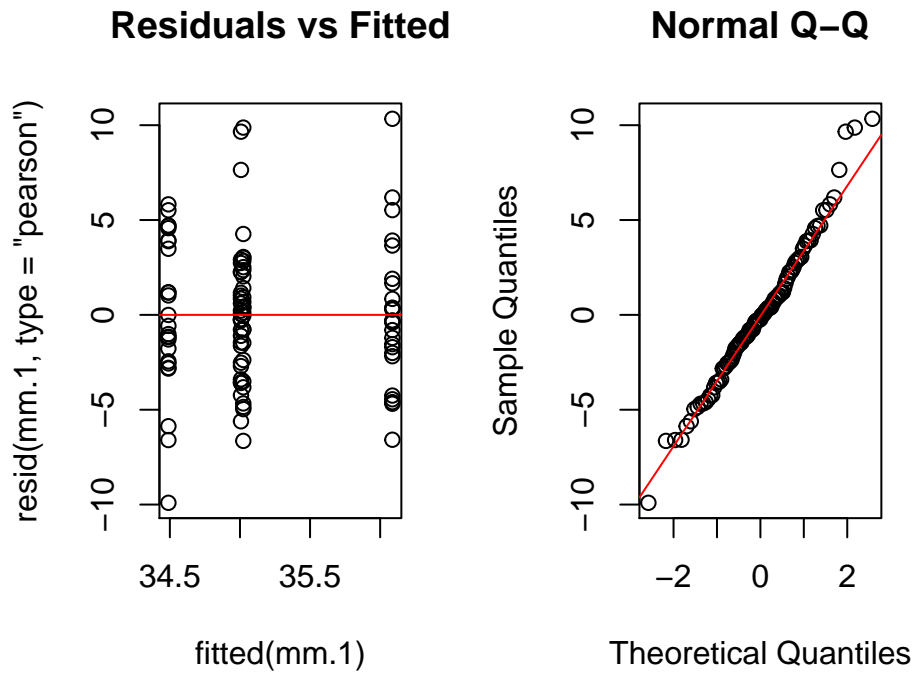
```
summary(mm.1)
```

```
##
## Call:
## lm(formula = CI_birth ~ Beach * Season, data = mum)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.9070 -2.3769 -0.2243  2.2525 10.3369
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    35.00728    0.72956  47.984  <2e-16 ***
## BeachSSB         1.08003    1.04202   1.036   0.303
## Season1920       0.01646    1.04202   0.016   0.987
## BeachSSB:Season1920 -1.61425    1.48084  -1.090   0.278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.72 on 97 degrees of freedom
## Multiple R-squared:  0.02456,    Adjusted R-squared:  -0.005609
## F-statistic: 0.8141 on 3 and 97 DF,  p-value: 0.4891
```

Check residuals

```
par(mfrow = c(1, 2))
# Check the linear relationship assumption. A horizontal line, without distinct
# patterns is an indication for a linear relationship.
plot(fitted(mm.1), resid(mm.1, type = "pearson"), main = "Residuals vs Fitted")
abline(0, 0, col = "red")

# Check if residuals are normally distributed. Points should follow the straight
# line.
qqnorm(resid(mm.1), main = "Normal Q-Q")
qqline(resid(mm.1), col = "red")
```

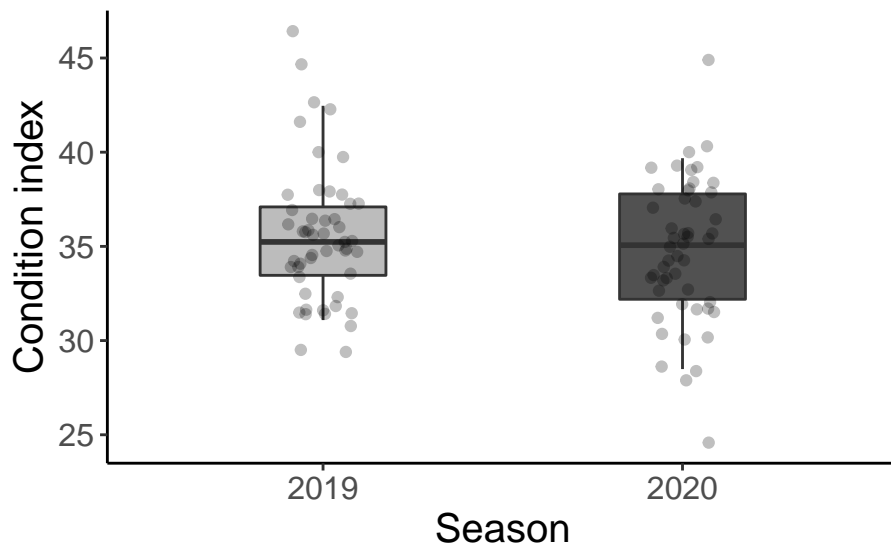
```
# Additional check for homogeneity of variance - Levene's Test
leveneTest(Weight_Birth ~ Beach * Season, data = mum)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 3  1.7622 0.1595
##      97
```

```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom = "boxplot", aes(y =
  CI_birth, x = Season, fill = Season), na.rm = TRUE, fill = c("#bcbcbc", "#515151"),
  width = 0.35) + geom_point(aes(y = CI_birth, x = Season, fill = Season), na.rm =
  TRUE, position = position_jitterdodge(), alpha = 0.25) + theme_classic() + theme(text
  = element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5,
  5), "mm"), plot.title = element_text(margin = margin(b = 0), size = 15),
  plot.subtitle = element_text(hjust = 0.28, margin = margin(t = 5, b = 10), size =
  12)) + labs(x = "Season", y = "Condition index", title = "Maternal condition
  postpartum", subtitle = expression(paste("estimate = 0.875, ", italic("p"), " =
  0.438"))) + scale_x_discrete(breaks = c("1819", "1920"), labels = c("2019", "2020"))
```

Maternal condition postpartum

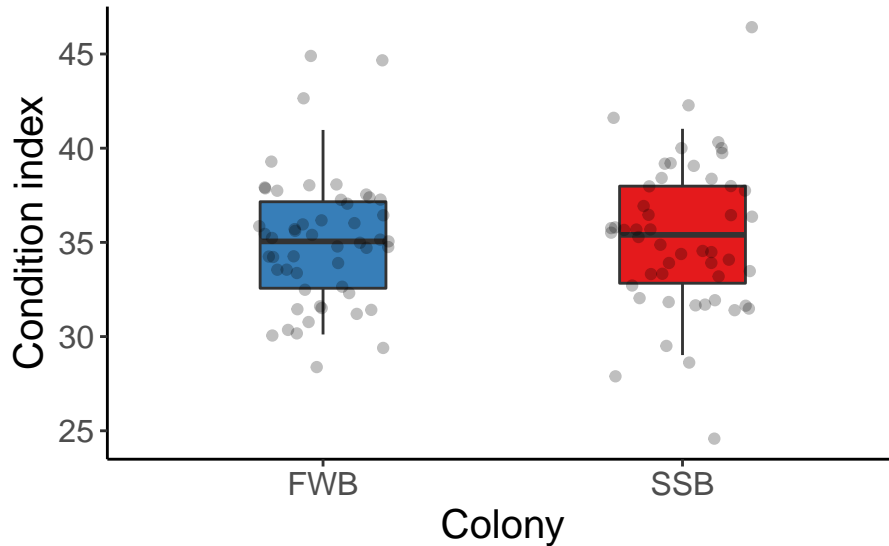
estimate = 0.875, $p = 0.438$



```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom = "boxplot", aes(y =  
  CI_birth, x = Beach, fill = Beach), na.rm = TRUE, fill = c("#377eb8", "#e41a1c"),  
  width = 0.35) + geom_point(aes(y = CI_birth, x = Beach, fill = Beach), na.rm = TRUE,  
  position = position_jitterdodge(), alpha = 0.25) + theme_classic() + theme(text =  
  element_text(size = 15), legend.position = "none", plot.margin = unit(c(10, 5, 5, 5),  
  "mm"), plot.title = element_text(margin = margin(b = 0), size = 15), plot.subtitle =  
  element_text(hjust = 0.28, margin = margin(t = 5, b = 10), size = 12)) + labs(x =  
  "Colony", y = "Condition index", title = "Maternal condition postpartum", subtitle =  
  expression(paste("estimate = 1.685, ", italic("p"), " = 0.137")))
```

Maternal condition postpartum

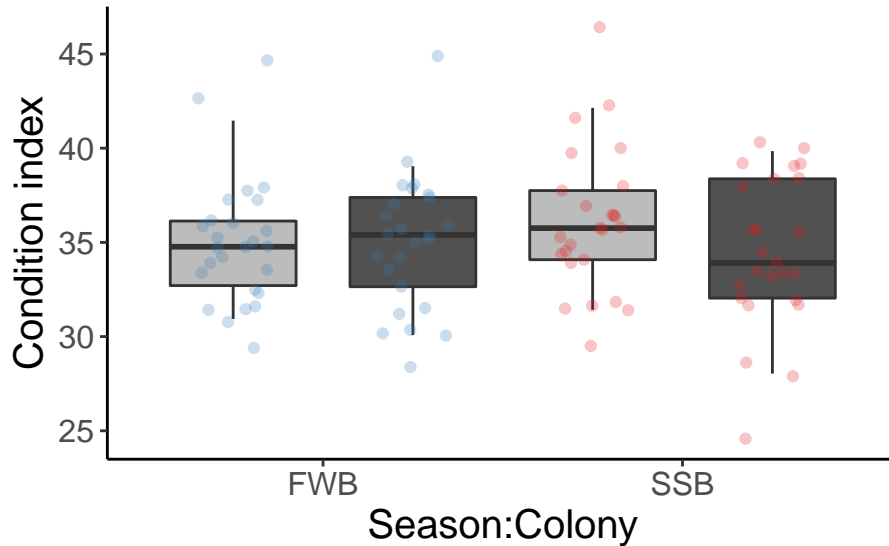
estimate = 1.685, $p = 0.137$



```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y =
  CI_birth, x = Beach, fill = Season), na.rm=TRUE, position=position_dodge(width=1),
  width = 0.7) + geom_point(aes(y = CI_birth, x = Beach, fill = Season, color = Beach),
  na.rm=TRUE, position = position_jitterdodge(dodge.width = 1), alpha = 0.25) +
  theme_classic() + theme(text = element_text(size=15), #legend.position = c(0.8, 1.1),
  legend.title = element_blank(), legend.position = "none",
  plot.margin=unit(c(10,5,5,5),"mm"), plot.title=element_text(margin=margin(b=0), size
  = 15), plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size =
  12)) + #guides(shape = FALSE, size = FALSE, fill = FALSE) + labs(x = "Season:Colony",
  y = "Condition index", title = "Maternal condition postpartum", subtitle =
  expression(paste("estimate = -2.497, ", italic("p"), " = 0.121"))) +
  scale_fill_manual(values=c("#bcbcbc", "#515151"), labels=c("2019", "2020"), name =
  "") + scale_color_manual(values=c("#377eb8", "#e41a1c"))
```

Maternal condition postpartum

estimate = -2.497, $p = 0.121$



Additional quality traits: weight, length, girth, span

Here we are interested in other quality traits including weight, length, girth, and span that may differ between focal mothers across colonies. Note, two length, girth, and span measurements were taken for each focal mother; these were averaged. Weight of the mother is weight in KG just after giving birth.

To compare trait values between colonies, we first check for normality

```
# weight  
with(mum, shapiro.test(Weight_Birth[Beach == "FWB"]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Weight_Birth[Beach == "FWB"]  
## W = 0.96939, p-value = 0.2087
```

```
with(mum, shapiro.test(Weight_Birth[Beach == "SSB"]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Weight_Birth[Beach == "SSB"]  
## W = 0.96058, p-value = 0.094
```

```
hist(mum$Weight_Birth)
```



```
# length  
with(mum, shapiro.test(Length[Beach == "FWB"]))
```

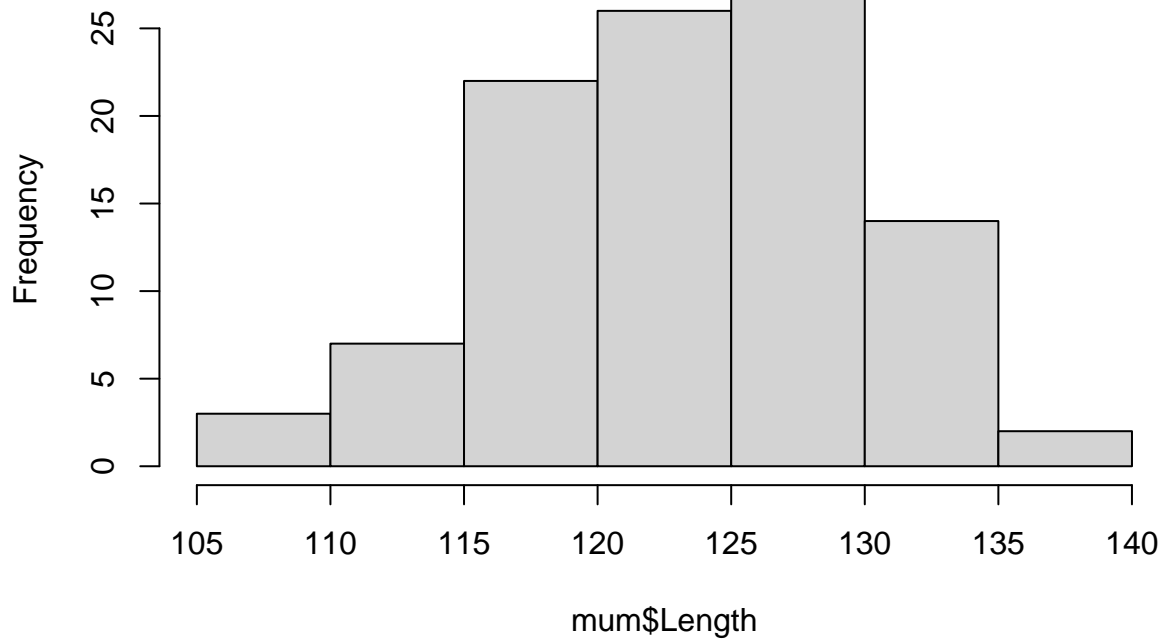
```
##  
## Shapiro-Wilk normality test  
##  
## data: Length[Beach == "FWB"]  
## W = 0.98846, p-value = 0.8982
```

```
with(mum, shapiro.test(Length[Beach == "SSB"]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Length[Beach == "SSB"]  
## W = 0.98586, p-value = 0.8079
```

```
hist(mum$Length)
```

Histogram of mum\$Length



```
# span  
with(mum, shapiro.test(Span[Beach == "FWB"]))
```

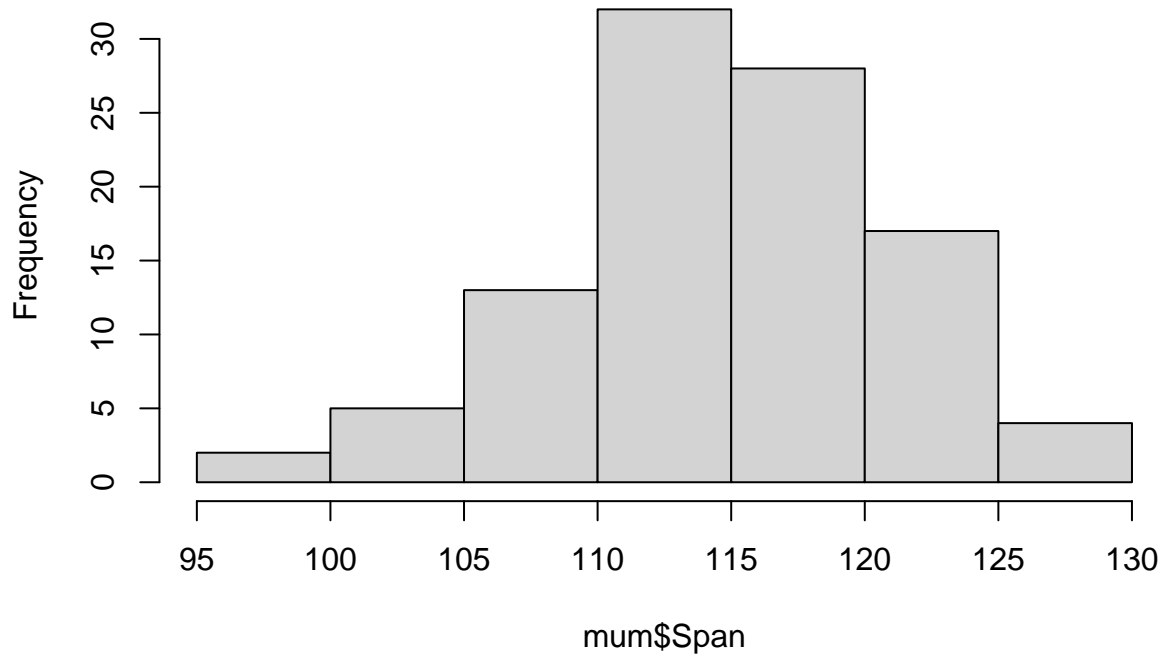
```
##  
## Shapiro-Wilk normality test  
##  
## data: Span[Beach == "FWB"]  
## W = 0.96449, p-value = 0.1296
```

```
with(mum, shapiro.test(Span[Beach == "SSB"]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Span[Beach == "SSB"]  
## W = 0.94931, p-value = 0.03197
```

```
hist(mum$Span)
```

Histogram of mum\$Span



```
# girth  
with(mum, shapiro.test(Girth[Beach == "FWB"]))
```

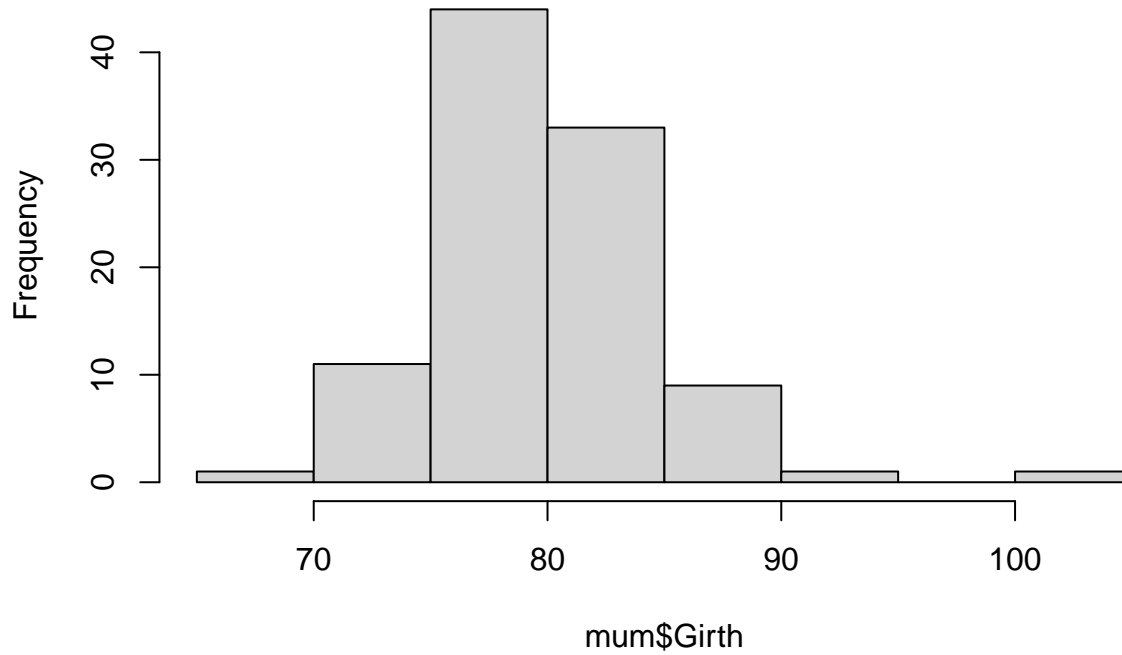
```
##  
## Shapiro-Wilk normality test  
##  
## data:  Girth[Beach == "FWB"]  
## W = 0.90329, p-value = 0.0006183
```

```
with(mum, shapiro.test(Girth[Beach == "SSB"]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  Girth[Beach == "SSB"]  
## W = 0.98867, p-value = 0.9104
```

```
hist(mum$Girth)
```

Histogram of mum\$Girth



```
# CI  
with(mum, shapiro.test(CI_birth[Beach == "FWB"]))
```

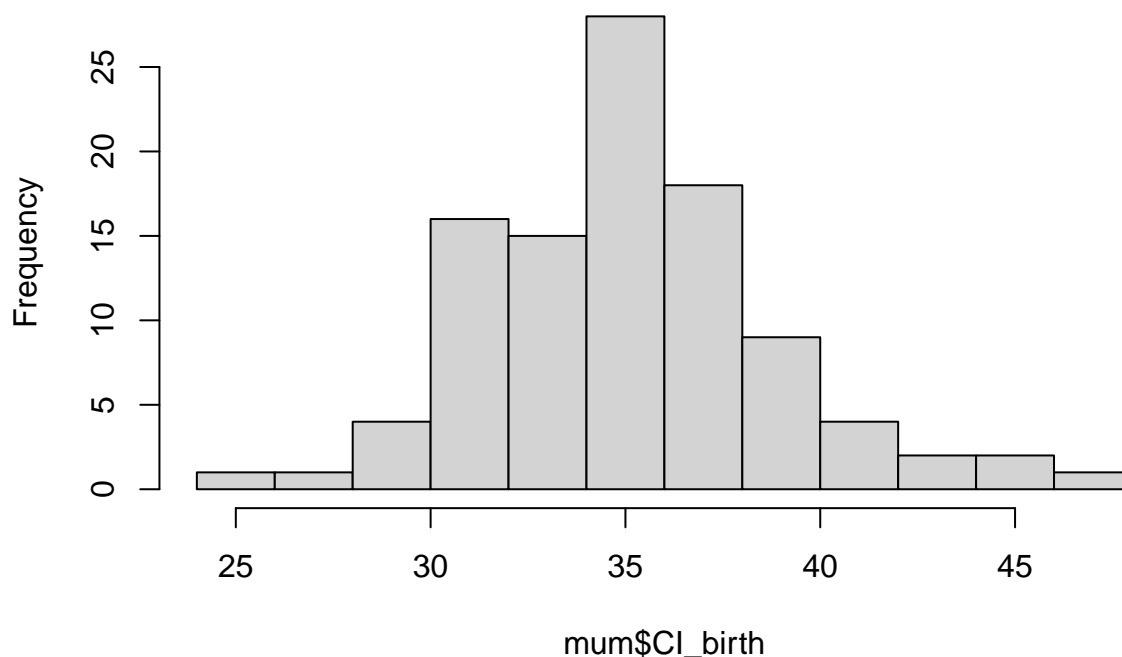
```
##  
## Shapiro-Wilk normality test  
##  
## data: CI_birth[Beach == "FWB"]  
## W = 0.94944, p-value = 0.02988
```

```
with(mum, shapiro.test(CI_birth[Beach == "SSB"]))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: CI_birth[Beach == "SSB"]  
## W = 0.98761, p-value = 0.8752
```

```
hist(mum$CI_birth)
```


Histogram of mum\$CI_birth



Given that many traits are not normally distributed, we use non parametric two-samples Wilcoxon rank tests to compare the two groups.

```
wilcox.test(Weight_Birth ~ Beach, data = mum, exact = FALSE)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: Weight_Birth by Beach  
## W = 1099.5, p-value = 0.2345  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Length ~ Beach, data = mum, exact = FALSE)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: Length by Beach  
## W = 1201.5, p-value = 0.6199  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Span ~ Beach, data = mum, exact = FALSE)
```

```
##
```

```
## Wilcoxon rank sum test with continuity correction
##
## data: Span by Beach
## W = 987.5, p-value = 0.05116
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Girth ~ Beach, data = mum, exact = FALSE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: Girth by Beach
## W = 1225.5, p-value = 0.8685
## alternative hypothesis: true location shift is not equal to 0
```

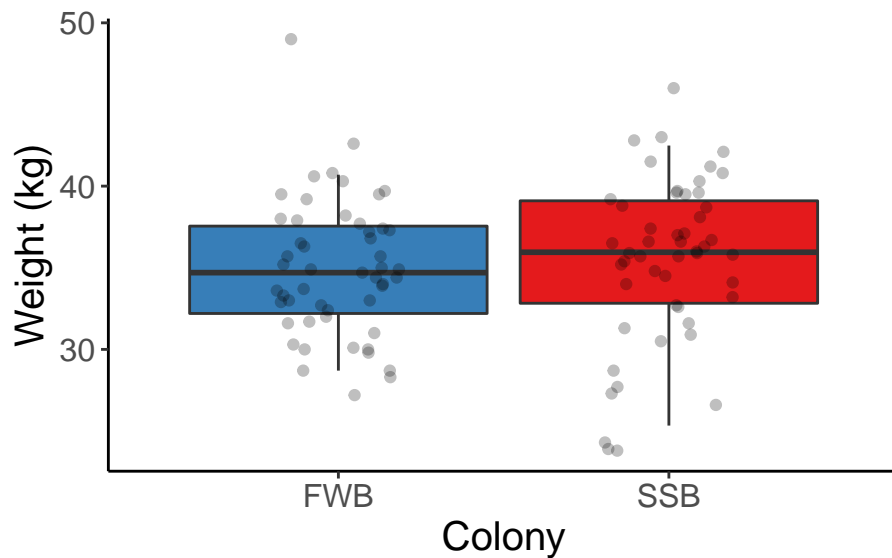
```
wilcox.test(CI_birth ~ Beach, data = mum, exact = FALSE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: CI_birth by Beach
## W = 1184, p-value = 0.5387
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y =
  Weight_Birth, x = Beach, fill = Beach), na.rm=TRUE, position=position_dodge(width=1))
+ geom_point(aes(y = Weight_Birth, x = Beach, fill = Beach), na.rm=TRUE, position =
  position_jitterdodge(dodge.width = 1), alpha = 0.25) + theme_classic() + theme(text =
  element_text(size=15), #legend.position = c(0.8, 1.1), legend.title =
  element_blank(), legend.position = "none", plot.margin=unit(c(10,5,5,5),"mm"),
  plot.title=element_text(margin=margin(b=0), size = 15),
  plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size = 12)) +
  #guides(shape = FALSE, size = FALSE, fill = FALSE) + labs(x = "Colony", y = "Weight
  (kg)", title = "(A) Postpartum weight", subtitle = expression(paste("W = 1099.5, ",
  italic("p"), " = 0.235"))) + scale_fill_manual(values=c("#377eb8", "#e41a1c"),
  labels=c("FWB", "SSB"), name = "")
```

(A) Postpartum weight

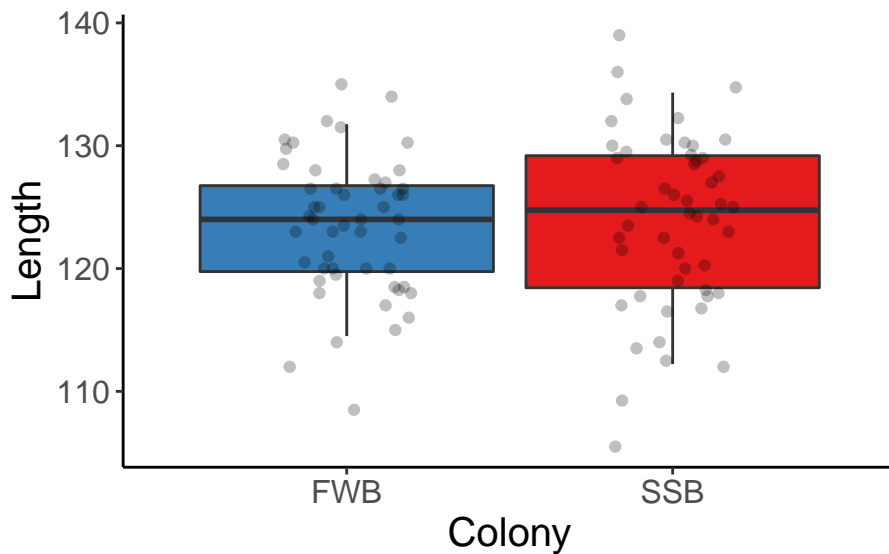
$W = 1099.5$, $p = 0.235$



```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y =  
  Length, x = Beach, fill = Beach), na.rm=TRUE, position=position_dodge(width=1)) +  
  geom_point(aes(y = Length, x = Beach, fill = Beach), na.rm=TRUE, position =  
  position_jitterdodge(dodge.width = 1), alpha = 0.25) + theme_classic() + theme(text =  
  element_text(size=15), #legend.position = c(0.8, 1.1), legend.title =  
  element_blank(), legend.position = "none", plot.margin=unit(c(10,5,5,5),"mm"),  
  plot.title=element_text(margin=margin(b=0), size = 15),  
  plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size = 12)) +  
  #guides(shape = FALSE, size = FALSE, fill = FALSE) + labs(x = "Colony", y = "Length",  
  title = "(B) Length", subtitle = expression(paste("W = 1201.5, ", italic("p"), " =  
  0.620"))) + scale_fill_manual(values=c("#377eb8", "#e41a1c"), labels=c("FWB", "SSB"),  
  name = "")
```

(B) Length

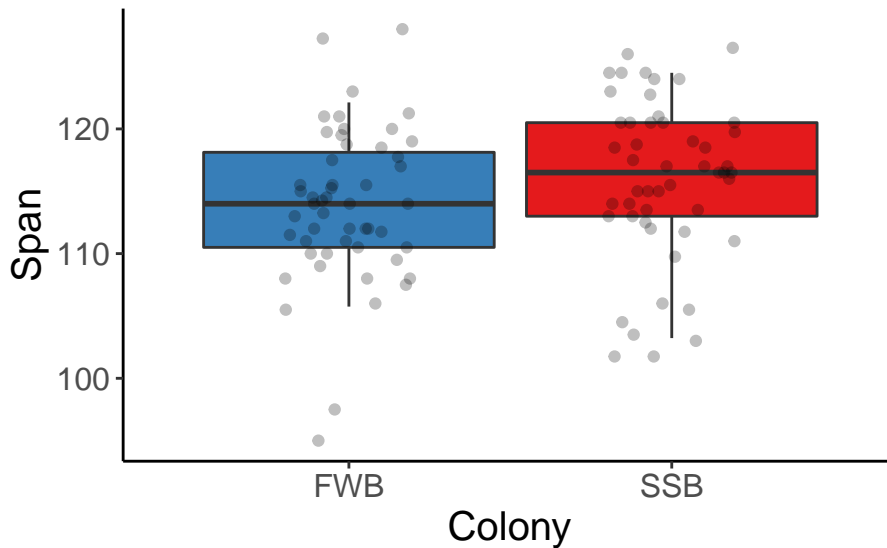
$W = 1201.5$, $p = 0.620$



```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y = Span,
  x = Beach, fill = Beach), na.rm=TRUE, position=position_dodge(width=1)) +
  geom_point(aes(y = Span, x = Beach, fill = Beach), na.rm=TRUE, position =
  position_jitterdodge(dodge.width = 1), alpha = 0.25) + theme_classic() + theme(text =
  element_text(size=15), #legend.position = c(0.8, 1.1), legend.title =
  element_blank(), legend.position = "none", plot.margin=unit(c(10,5,5,5),"mm"),
  plot.title=element_text(margin=margin(b=0), size = 15),
  plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size = 12)) +
  #guides(shape = FALSE, size = FALSE, fill = FALSE) + labs(x = "Colony", y = "Span",
  title = "(C) Span", subtitle = expression(paste("W = 987.5, ", italic("p"), " =
  0.051"))) + scale_fill_manual(values=c("#377eb8", "#e41a1c"), labels=c("FWB", "SSB"),
  name = "")
```

(C) Span

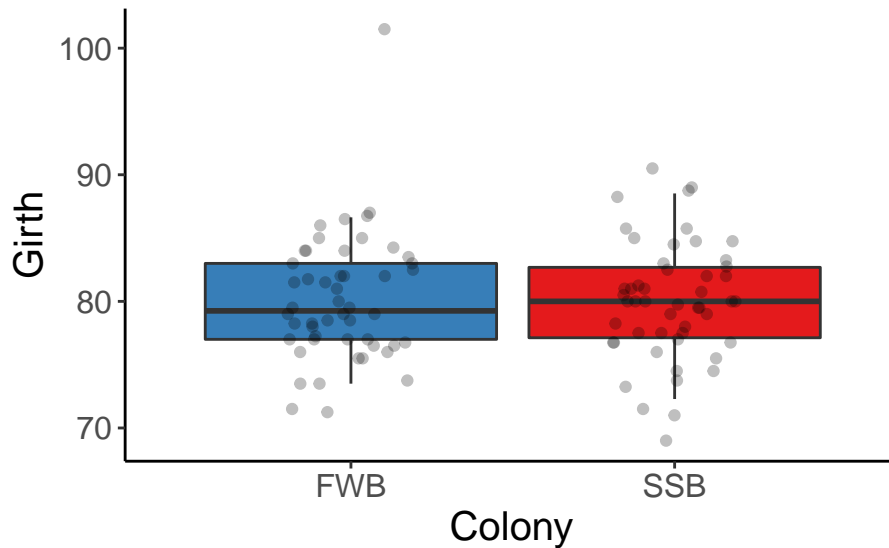
$W = 987.5$, $p = 0.051$



```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y = Girth,
  x = Beach, fill = Beach), na.rm=TRUE, position=position_dodge(width=1)) +
  geom_point(aes(y = Girth, x = Beach, fill = Beach), na.rm=TRUE, position =
  position_jitterdodge(dodge.width = 1), alpha = 0.25) + theme_classic() + theme(text =
  element_text(size=15), #legend.position = c(0.8, 1.1), legend.title =
  element_blank(), legend.position = "none", plot.margin=unit(c(10,5,5,5),"mm"),
  plot.title=element_text(margin=margin(b=0), size = 15),
  plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size = 12)) +
  #guides(shape = FALSE, size = FALSE, fill = FALSE) + labs(x = "Colony", y = "Girth",
  title = "(D) Girth", subtitle = expression(paste("W = 1225.5, ", italic("p"), " =
  0.869"))) + scale_fill_manual(values=c("#377eb8", "#e41a1c"), labels=c("FWB", "SSB"),
  name = "")
```

(D) Girth

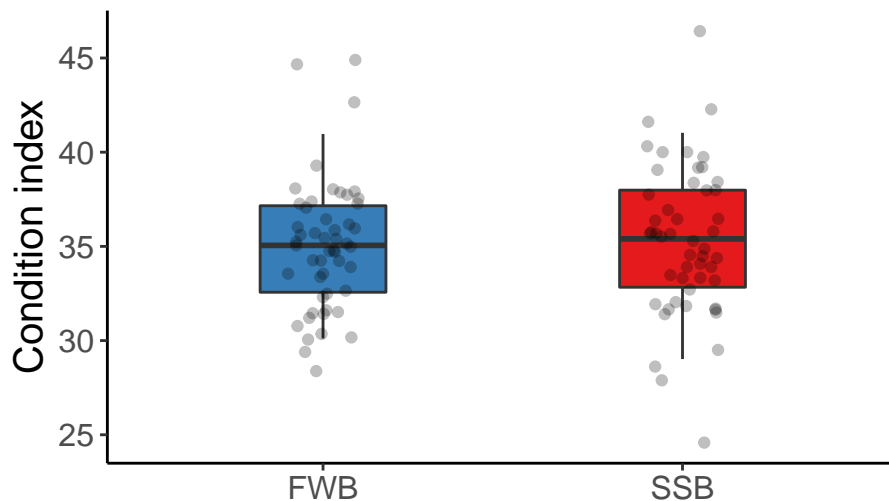
$W = 1225.5$, $p = 0.869$



```
ggplot(data = mum) + stat_summary(fun.data = quantiles_95, geom="boxplot", aes(y =  
  CI_birch, x = Beach, fill = Beach), na.rm=TRUE, #position=position_dodge(width=1),  
  width = 0.35) + geom_point(aes(y = CI_birch, x = Beach, fill = Beach), na.rm=TRUE,  
  position = position_jitterdodge(jitter.width = 0.2), alpha = 0.25) + theme_classic()  
  + theme(text = element_text(size=15), #legend.position = c(0.8, 1.1), legend.title =  
  element_blank(), legend.position = "none", plot.margin=unit(c(10,5,5,5),"mm"),  
  plot.title=element_text(margin=margin(b=0), size = 15),  
  plot.subtitle=element_text(hjust = 0.28, margin=margin(t=5, b = 10), size = 12)) +  
  #guides(shape = FALSE, size = FALSE, fill = FALSE) + labs(x = "", y = "Condition  
  index", title = "(E) Condition index", subtitle = expression(paste("W = 1184, ",  
  italic("p"), " = 0.539"))) + scale_fill_manual(values=c("#377eb8", "#e41a1c"),  
  labels=c("FWB", "SSB"), name = "")
```

(E) Condition index

W = 1184, $p = 0.539$



```
## - Session info -----
## setting value
## version R version 4.0.2 (2020-06-22)
## os Windows 10 x64
## system x86_64, mingw32
## ui RTerm
## language (EN)
## collate English_United States.1252
## ctype English_United States.1252
## tz Europe/Berlin
## date 2021-02-21
##
## - Packages -----
## package * version date lib source
## abind 1.4-5 2016-07-21 [1] CRAN (R 4.0.0)
## assertthat 0.2.1 2019-03-21 [1] CRAN (R 4.0.2)
## backports 1.1.10 2020-09-15 [1] CRAN (R 4.0.2)
## bayestestR 0.7.2 2020-07-20 [1] CRAN (R 4.0.2)
## bbmle * 1.0.23.1 2020-02-03 [1] CRAN (R 4.0.2)
## bdsmatrix 1.3-4 2020-01-13 [1] CRAN (R 4.0.0)
## blob 1.2.1 2020-01-20 [1] CRAN (R 4.0.2)
## boot 1.3-25 2020-04-26 [1] CRAN (R 4.0.3)
## broom 0.7.0 2020-07-09 [1] CRAN (R 4.0.2)
## car * 3.0-9 2020-08-11 [1] CRAN (R 4.0.2)
## carData * 3.0-4 2020-05-22 [1] CRAN (R 4.0.0)
## cellranger 1.1.0 2016-07-27 [1] CRAN (R 4.0.2)
## class 7.3-17 2020-04-26 [2] CRAN (R 4.0.2)
## cli 2.0.2 2020-02-28 [1] CRAN (R 4.0.2)
## coda 0.19-3 2019-07-05 [1] CRAN (R 4.0.2)
```

```

## codetools      0.2-16      2018-12-24 [2] CRAN (R 4.0.2)
## coin          1.3-1       2019-08-28 [1] CRAN (R 4.0.2)
## colorspace    1.4-1       2019-03-18 [1] CRAN (R 4.0.2)
## cowplot       1.1.0       2020-09-08 [1] CRAN (R 4.0.2)
## crayon        1.3.4       2017-09-16 [1] CRAN (R 4.0.2)
## curl          4.3         2019-12-02 [1] CRAN (R 4.0.2)
## data.table    1.13.0      2020-07-24 [1] CRAN (R 4.0.2)
## DBI           1.1.0       2019-12-15 [1] CRAN (R 4.0.2)
## dbplyr        1.4.4       2020-05-27 [1] CRAN (R 4.0.2)
## DescTools     0.99.38     2020-09-07 [1] CRAN (R 4.0.2)
## digest        0.6.25     2020-02-23 [1] CRAN (R 4.0.2)
## dplyr         * 1.0.2     2020-08-18 [1] CRAN (R 4.0.2)
## e1071         1.7-3       2019-11-26 [1] CRAN (R 4.0.2)
## effectsize    0.3.2       2020-07-27 [1] CRAN (R 4.0.2)
## ellipsis      0.3.1       2020-05-15 [1] CRAN (R 4.0.2)
## emmeans      * 1.5.0     2020-08-18 [1] CRAN (R 4.0.2)
## EMT           1.1         2013-01-29 [1] CRAN (R 4.0.0)
## estimability  1.3         2018-02-11 [1] CRAN (R 4.0.0)
## evaluate      0.14        2019-05-28 [1] CRAN (R 4.0.2)
## Exact         2.0         2019-10-14 [1] CRAN (R 4.0.0)
## expm          0.999-5     2020-07-20 [1] CRAN (R 4.0.2)
## extrafont     0.17        2014-12-08 [1] CRAN (R 4.0.0)
## extrafontdb   1.0         2012-06-11 [1] CRAN (R 4.0.0)
## fansi         0.4.1       2020-01-08 [1] CRAN (R 4.0.2)
## farver        2.0.3       2020-01-16 [1] CRAN (R 4.0.2)
## forcats      * 0.5.0     2020-03-01 [1] CRAN (R 4.0.2)
## foreign       0.8-80      2020-05-24 [2] CRAN (R 4.0.2)
## formatR      * 1.7        2019-06-11 [1] CRAN (R 4.0.2)
## fs            1.5.0       2020-07-31 [1] CRAN (R 4.0.2)
## generics      0.0.2       2018-11-29 [1] CRAN (R 4.0.2)
## ggeffects     0.16.0     2020-09-13 [1] CRAN (R 4.0.2)
## ggplot2      * 3.3.2     2020-06-19 [1] CRAN (R 4.0.2)
## ggpubr       * 0.4.0     2020-06-27 [1] CRAN (R 4.0.2)
## ggsignif      0.6.0       2019-08-08 [1] CRAN (R 4.0.2)
## gld           2.6.2       2020-01-08 [1] CRAN (R 4.0.2)
## glue          1.4.2       2020-08-27 [1] CRAN (R 4.0.2)
## gridExtra     2.3         2017-09-09 [1] CRAN (R 4.0.2)
## gtable        0.3.0       2019-03-25 [1] CRAN (R 4.0.2)
## haven         2.3.1       2020-06-01 [1] CRAN (R 4.0.2)
## hms           0.5.3       2020-01-08 [1] CRAN (R 4.0.2)
## htmltools     0.5.0       2020-06-16 [1] CRAN (R 4.0.2)
## httr          1.4.2       2020-07-20 [1] CRAN (R 4.0.2)
## insight       0.9.5       2020-09-07 [1] CRAN (R 4.0.2)
## jsonlite      1.7.1       2020-09-07 [1] CRAN (R 4.0.2)
## kableExtra    * 1.2.1     2020-08-27 [1] CRAN (R 4.0.2)
## knitr         * 1.29      2020-06-23 [1] CRAN (R 4.0.2)
## labeling      0.3         2014-08-23 [1] CRAN (R 4.0.0)
## lattice      * 0.20-41   2020-04-02 [2] CRAN (R 4.0.2)
## libcoin       1.0-6       2020-08-14 [1] CRAN (R 4.0.2)
## lifecycle     0.2.0       2020-03-06 [1] CRAN (R 4.0.2)
## lme4          * 1.1-23    2020-04-07 [1] CRAN (R 4.0.2)
## lmerTest      * 3.1-2     2020-04-08 [1] CRAN (R 4.0.2)
## lmom          2.8         2019-03-12 [1] CRAN (R 4.0.0)
## lmtest        0.9-38     2020-09-09 [1] CRAN (R 4.0.2)

```



```

## lsmeans      * 2.30-0      2018-11-02 [1] CRAN (R 4.0.2)
## lubridate    1.7.9        2020-06-08 [1] CRAN (R 4.0.2)
## magrittr     1.5          2014-11-22 [1] CRAN (R 4.0.2)
## MASS         7.3-52       2020-08-18 [2] CRAN (R 4.0.2)
## Matrix       * 1.2-18       2019-11-27 [2] CRAN (R 4.0.2)
## matrixStats  0.56.0        2020-03-13 [1] CRAN (R 4.0.2)
## mgcv         1.8-33        2020-08-27 [2] CRAN (R 4.0.2)
## minqa        1.2.4         2014-10-09 [1] CRAN (R 4.0.2)
## modelr       0.1.8         2020-05-19 [1] CRAN (R 4.0.2)
## modeltools   0.2-23        2020-03-05 [1] CRAN (R 4.0.0)
## multcomp     1.4-13        2020-04-08 [1] CRAN (R 4.0.2)
## multcompView 0.1-8         2019-12-19 [1] CRAN (R 4.0.2)
## munsell      0.5.0         2018-06-12 [1] CRAN (R 4.0.2)
## mvtnorm      1.1-1         2020-06-09 [1] CRAN (R 4.0.0)
## nlme         3.1-149       2020-08-23 [2] CRAN (R 4.0.2)
## nloptr       1.2.2.2       2020-07-02 [1] CRAN (R 4.0.2)
## nortest      1.0-4         2015-07-30 [1] CRAN (R 4.0.0)
## numDeriv     2016.8-1.1    2019-06-06 [1] CRAN (R 4.0.0)
## openxlsx     4.1.5         2020-05-06 [1] CRAN (R 4.0.2)
## parameters   0.8.5         2020-09-12 [1] CRAN (R 4.0.2)
## pbkrtest     0.4-8.6       2020-02-20 [1] CRAN (R 4.0.2)
## performance  0.5.0         2020-09-12 [1] CRAN (R 4.0.2)
## pillar       1.4.6         2020-07-10 [1] CRAN (R 4.0.2)
## pkgconfig    2.0.3         2019-09-22 [1] CRAN (R 4.0.2)
## plyr         1.8.6         2020-03-03 [1] CRAN (R 4.0.2)
## purrr        * 0.3.4        2020-04-17 [1] CRAN (R 4.0.2)
## R6           2.4.1         2019-11-12 [1] CRAN (R 4.0.2)
## RColorBrewer 1.1-2         2014-12-07 [1] CRAN (R 4.0.0)
## rcompanion   * 2.3.25       2020-02-09 [1] CRAN (R 4.0.2)
## Rcpp         1.0.5         2020-07-06 [1] CRAN (R 4.0.2)
## readr        * 1.3.1        2018-12-21 [1] CRAN (R 4.0.2)
## readxl       1.3.1         2019-03-13 [1] CRAN (R 4.0.2)
## reprex       0.3.0         2019-05-16 [1] CRAN (R 4.0.2)
## rio          0.5.16        2018-11-26 [1] CRAN (R 4.0.2)
## rlang        0.4.8         2020-10-08 [1] CRAN (R 4.0.3)
## rmarkdown    2.3           2020-06-18 [1] CRAN (R 4.0.2)
## rstatix      0.6.0         2020-06-18 [1] CRAN (R 4.0.2)
## rstudioapi   0.11          2020-02-07 [1] CRAN (R 4.0.2)
## Rttf2pt1     1.3.8         2020-01-10 [1] CRAN (R 4.0.0)
## rvest        0.3.6         2020-07-25 [1] CRAN (R 4.0.2)
## sandwich     2.5-1         2019-04-06 [1] CRAN (R 4.0.2)
## scales       1.1.1         2020-05-11 [1] CRAN (R 4.0.2)
## sessioninfo  1.1.1         2018-11-05 [1] CRAN (R 4.0.2)
## sjlabelled   1.1.6         2020-06-25 [1] CRAN (R 4.0.2)
## sjmisc       2.8.5         2020-05-28 [1] CRAN (R 4.0.2)
## sjPlot       * 2.8.4        2020-05-24 [1] CRAN (R 4.0.2)
## sjstats      0.18.0        2020-05-06 [1] CRAN (R 4.0.2)
## statmod      1.4.34        2020-02-17 [1] CRAN (R 4.0.2)
## stringi      1.5.3         2020-09-09 [1] CRAN (R 4.0.2)
## stringr      * 1.4.0        2019-02-10 [1] CRAN (R 4.0.2)
## survival     3.2-3         2020-06-13 [2] CRAN (R 4.0.2)
## TH.data      1.0-10        2019-01-21 [1] CRAN (R 4.0.2)
## tibble       * 3.0.3        2020-07-10 [1] CRAN (R 4.0.2)
## tidyr        * 1.1.2        2020-08-27 [1] CRAN (R 4.0.2)

```

```
## tidyselect      1.1.0      2020-05-11 [1] CRAN (R 4.0.2)
## tidyverse      * 1.3.0      2019-11-21 [1] CRAN (R 4.0.2)
## vcd            * 1.4-7      2020-04-02 [1] CRAN (R 4.0.2)
## vctr          0.3.4      2020-08-29 [1] CRAN (R 4.0.2)
## viridisLite    0.3.0      2018-02-01 [1] CRAN (R 4.0.2)
## waffle        * 0.7.0      2017-01-07 [1] CRAN (R 4.0.2)
## webshot        0.5.2      2019-11-22 [1] CRAN (R 4.0.2)
## withr         2.3.0      2020-09-22 [1] CRAN (R 4.0.2)
## xfun           0.17       2020-09-09 [1] CRAN (R 4.0.2)
## xml2           1.3.2      2020-04-23 [1] CRAN (R 4.0.2)
## xtable         1.8-4      2019-04-21 [1] CRAN (R 4.0.2)
## yaml           2.2.1      2020-02-01 [1] CRAN (R 4.0.2)
## zip            2.1.1      2020-08-27 [1] CRAN (R 4.0.2)
## zoo            1.8-8      2020-05-02 [1] CRAN (R 4.0.2)
##
## [1] C:/Users/localadmin/Documents/R/win-library/4.0
## [2] C:/Program Files/R/R-4.0.2/library
```