11_survival_model.R

dahliafoo

2022-06-18

```
# survival model

rm(list = ls())
options(tibble.width = Inf)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(sf)
```

```
## Linking to GEOS 3.8.1, GDAL 3.2.1, PROJ 7.2.1
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(foieGras)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(tmap)
```

```
## Registered S3 methods overwritten by 'stars':
##   method             from
##   st_bbox.SpatRaster sf
##   st_crs.SpatRaster  sf
```

```
library(ggpubr)
library(see)
```

```
## Registered S3 methods overwritten by 'parameters':
##   method                           from
##   as.double.parameters_kurtosis    datawizard
##   as.double.parameters_skewness    datawizard
##   as.double.parameters_smoothness  datawizard
##   as.numeric.parameters_kurtosis   datawizard
##   as.numeric.parameters_skewness   datawizard
##   as.numeric.parameters_smoothness datawizard
##   print.parameters_distribution    datawizard
##   print.parameters_kurtosis        datawizard
##   print.parameters_skewness        datawizard
##   summary.parameters_kurtosis      datawizard
##   summary.parameters_skewness      datawizard
```

```
library(orsifronts)
```

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

```
library(pals)
```

```
##
## Attaching package: 'pals'
```

```
## The following objects are masked from 'package:viridis':
##
##     cividis, inferno, magma, plasma, turbo, viridis
```

```
## The following objects are masked from 'package:viridisLite':
##
##     cividis, inferno, magma, plasma, turbo, viridis
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(DHARMa)
```

```
## This is DHARMa 0.4.3. For overview type ’?DHARMa’. For recent changes, type news(package = ’DHARMa’)
```

```
library(MuMIn)
library(mgcv)
```

```
## Loading required package: nlme

##
## Attaching package: ’nlme’

## The following object is masked from ’package:lme4’:
##
##     lmList

## The following object is masked from ’package:dplyr’:
##
##     collapse

## This is mgcv 1.8-36. For overview type ’help("mgcv-package")’.
```

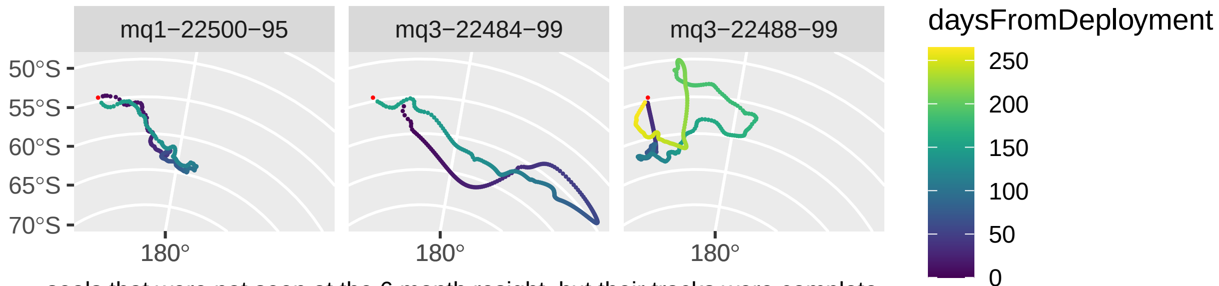
```
library(effects)
```

```
## Loading required package: carData

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(ggeffects)
# library(sjPlot)

load('11_survival_model_workspace.RData')


# Load Datasets --------------------------------------------------------------

source('convert2polarsf.R')
load('baseInfo.Rdata')
source("functions.R")
d1 <- readRDS("./Output/all_data_combined.rds")
# d1 <- readRDS("./Output/all_data_combined__REF_DATE=6d.rds")
dr <- d1 %>% filter(sim == 0, trip == 1, land == FALSE, !is.na(bearing.pt)) # only real locations

# Determine if seal swimming with the current -----------------------------
dr$compass_zone.pt <- sapply(dr$bearing.pt, whichZone)

# Some exploratory plots ---------------------------------------------------
# look at tracks of seals that wasn't seen again after 6m but track was completed.
tmp <- dr %>%
  filter(seen_6m == FALSE, is_trip_complete == TRUE)

tmp <- convert2polarsf(tmp)
```

```
## Loading required package: raster

##
## Attaching package: 'raster'

## The following object is masked from 'package:nlme':
##
##     getData

## The following object is masked from 'package:lme4':
##
##     getData

## The following object is masked from 'package:dplyr':
##
##     select
```

```
tmp %>% ggplot() +
  geom_sf(aes(col = daysFromDeployment), size = .1) +
  geom_sf(data = convert2polarsf(mq), col = 'red', size = 0.1) +
  facet_wrap(~id) +
  scale_color_viridis() +
  labs(caption = "seals that were not seen at the 6 month resight, but their tracks were complete.")
```



seals that were not seen at the 6 month resight, but their tracks were complete.

```
# Model -------------------------------------------------------------------
# + 1st trip survival -----------------------------------------------------
## prepare model data
dat <-dr %>%
  filter(trip == 1, land == FALSE) %>%
  group_by(id) %>%
  arrange(id, trip) %>%
  summarise(
    surviveTrip1 = ifelse(first(seen_6m) == TRUE | first(is_trip_complete) == TRUE, 1, 0),
    surviveYear1 = ifelse(first(seen_1y) == TRUE, 1, 0),
    # across(c(topo:ice, chl, chlgrad, dist_to_ice_m), mean, na.rm = T),
    across(c(blackmass, weanmass, bearing_diff, ew_zone, compass_zone, compass_zone.pt), first),
    # g_sd = sd(g),
    # g_mean = mean(g),
    # g_propARS = sum(g<=0.5)/n(),
    is_ESE = ifelse(compass_zone == "E-SE", TRUE, FALSE))

dat <- dat %>% filter(!is.na(weanmass), !is.na(is_ESE))
# dat <- dat %>% mutate(across(topo:bearing_diff, ~scale(.)[,1]))

# seals that survived 1st trip
table(dat$surviveTrip1 == TRUE)
```

```
##
## FALSE  TRUE
##    16    28
```

```
options(na.action = 'na.omit')

m.global <- glm(surviveTrip1 ~ weanmass*is_ESE, data = dat, family = binomial)
options(na.action = 'na.fail')
dd <- dredge(m.global)
```

```
## Fixed term is "(Intercept)"
```

```
dd
```

```
## Global model call: glm(formula = surviveTrip1 ~ weanmass * is_ESE, family = binomial,
##     data = dat)
## ---
## Model selection table
##     (Int) is_ESE       wnm is_ESE:wnm df  logLik AICc delta weight
## 2 -0.1178      +                       2 -27.206 58.7  0.00  0.435
## 1  0.5596                              1 -28.841 59.8  1.07  0.255
## 4  0.1839      + -0.002586             3 -27.163 60.9  2.22  0.143
## 3  0.7587        -0.001672             2 -28.822 61.9  3.23  0.087
## 8 -1.3460      +  0.010510          +  4 -26.531 62.1  3.38  0.080
## Models ranked by AICc(x)
```

```
# write.csv(dd %>% mutate(across(where(is.numeric), ~signif(., 3))),
#           "./output/surivalTrip1__dredge__model_selection.csv", na = "")
```

```
avgmod.2delta <- model.avg(dd, subset = delta < 2, fit = TRUE)
summary(avgmod.2delta)
```

```
##
## Call:
## model.avg(object = get.models(object = dd, subset = delta < 2))
##
## Component model call:
## glm(formula = surviveTrip1 ~ <2 unique rhs>, family = binomial, data =
##      dat)
##
## Component models:
##        df logLik  AICc delta weight
## 1       2 -27.21 58.70  0.00   0.63
## (Null)  1 -28.84 59.78  1.07   0.37
##
## Term codes:
## is_ESE
##      1
##
## Model-averaged coefficients:
## (full average)
##             Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)   0.1321     0.5404      0.5506   0.240    0.810
## is_ESETRUE    0.7368     0.7669      0.7774   0.948    0.343
##
## (conditional average)
##             Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)   0.1321     0.5404      0.5506   0.240   0.8104
## is_ESETRUE    1.1676     0.6550      0.6744   1.731   0.0834 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
importance(avgmod.2delta)
```

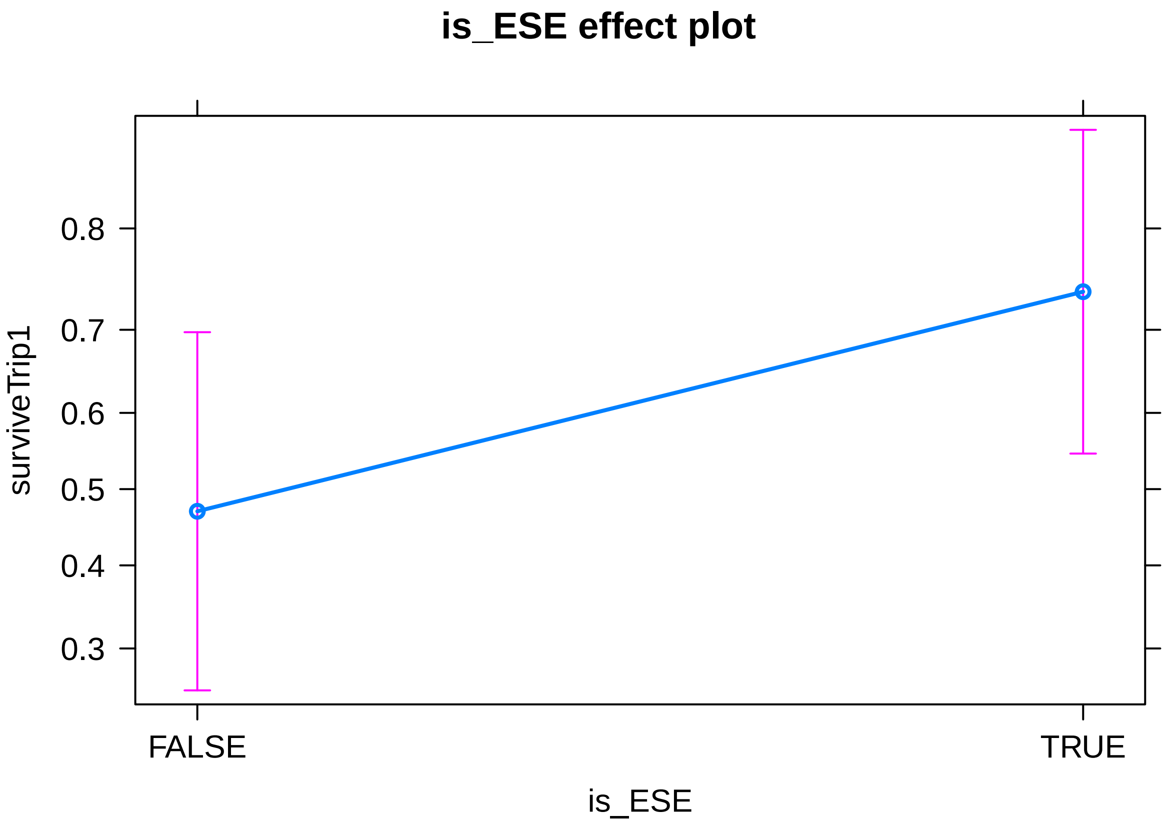
```
##                      is_ESE
## Sum of weights:      0.63
## N containing models:    1
```

```
# save model summary
# write.csv(summary(avgmod.2delta)$coefmat.subset %>% data.frame(), "./output/1st trip survival model t

m.final <- glm(surviveTrip1 ~  is_ESE, data = dat, family = binomial)
summary(m.final)
```

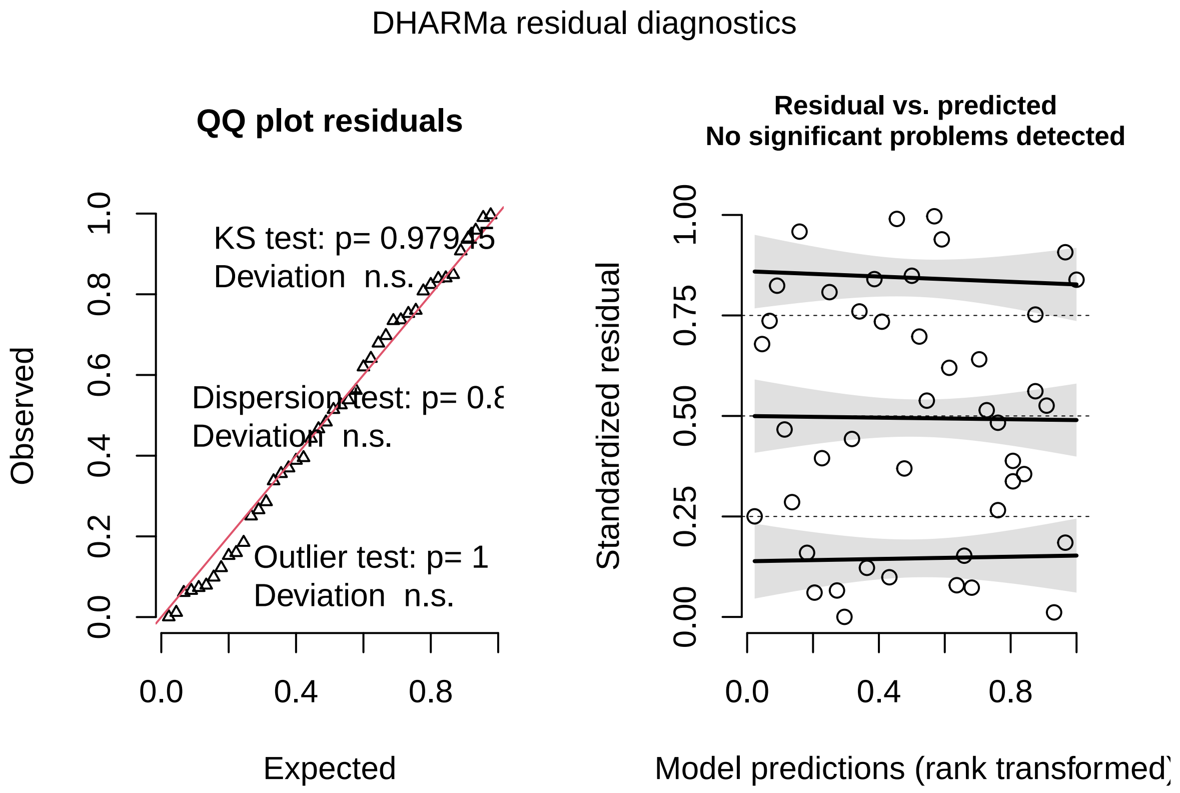
```
##
## Call:
## glm(formula = surviveTrip1 ~ is_ESE, family = binomial, data = dat)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6431  -1.1278   0.7747   0.7747   1.2278
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1178     0.4859  -0.242   0.8085
## is_ESETRUE    1.1676     0.6550   1.783   0.0746 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 57.682  on 43  degrees of freedom
## Residual deviance: 54.411  on 42  degrees of freedom
## AIC: 58.411
##
## Number of Fisher Scoring iterations: 4
```

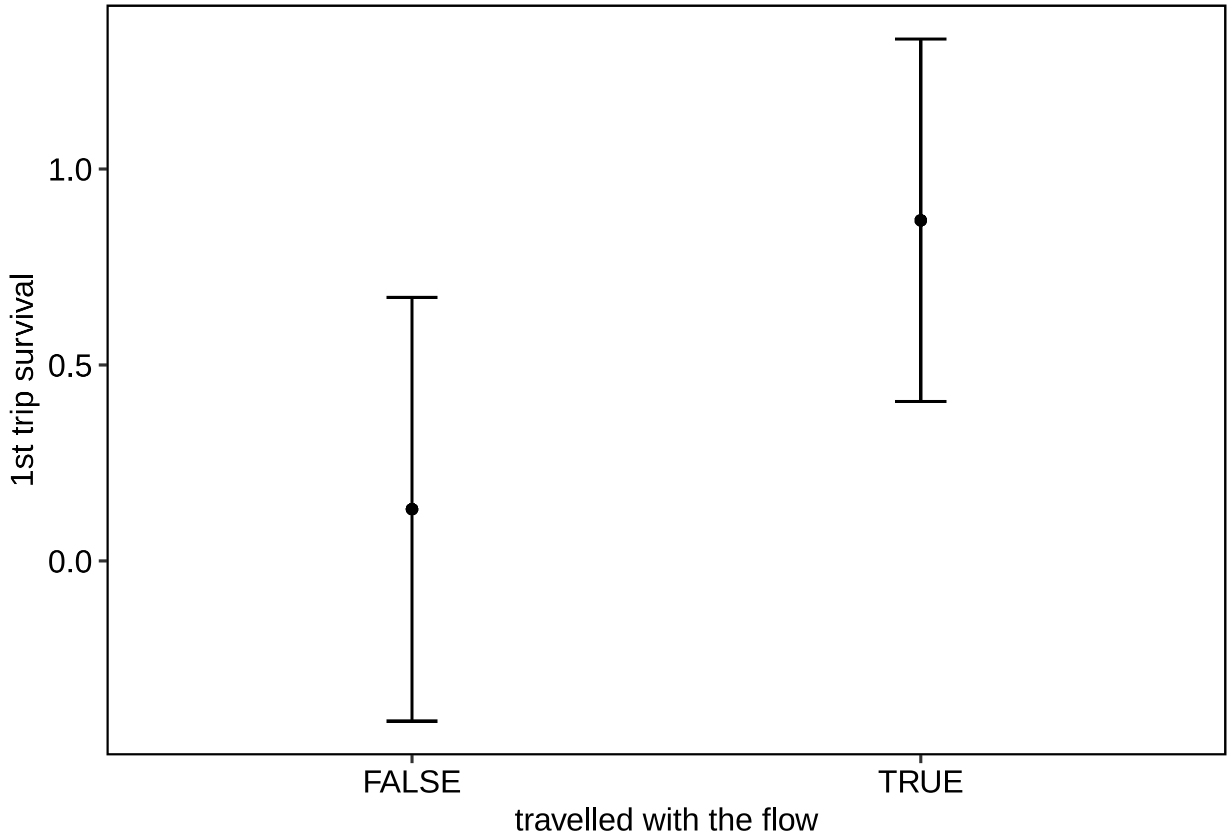
```
effects::allEffects(m.final) %>% plot()
```



```
simulationOutput <- simulateResiduals(fittedModel = m.global, n = 1000)
plot(simulationOutput)
```



```
# ~ plot avg model ---------------------------------------------------------
a <- tibble(is_ESE = TRUE, weanmass = dat$weanmass)
b <- tibble(is_ESE = FALSE, weanmass = dat$weanmass)

pred <- predict(avgmod.2delta, newdata = a, se.fit = TRUE) %>% as_tibble() %>%
  bind_cols(tibble(is_ESE = TRUE, weanmass = dat$weanmass))

pred <- pred %>% bind_rows(
  predict(avgmod.2delta, newdata = b, se.fit = TRUE) %>% as_tibble() %>% bind_cols(tibble(is_ESE = FALS
)

# png('./output/survival model/1st trip survival -- avg model effects.png', width=15, height=15, units=
  ggplot(data = pred, aes(x = is_ESE, group = is_ESE)) +
    geom_point(aes(y = fit)) +
    geom_errorbar(aes(ymin = fit-se.fit, ymax = fit+se.fit), width = .1) +
    labs(y = "1st trip survival", x = "travelled with the flow") +
    theme_pubr(border = T)
```



```
# dev.off()



# + 1st year survival -----------------------------------------------------
# seals that survived 1st year
table(dat$surviveYear1 == TRUE)
```

```
##
## FALSE  TRUE
##    28    16
```

```
options(na.action = 'na.omit')

m.global <- glm(surviveYear1 ~ weanmass*is_ESE, data = dat, family = binomial)
options(na.action = 'na.fail')
dd <- dredge(m.global)
```

```
## Fixed term is "(Intercept)"
```

```
dd
```

```
## Global model call: glm(formula = surviveYear1 ~ weanmass * is_ESE, family = binomial,
##     data = dat)
## ---
## Model selection table
##     (Int) is_ESE     wnm is_ESE:wnm df  logLik AICc delta weight
## 3 -2.3670        0.01492             2 -27.337 59.0  0.00  0.280
## 4 -2.9630      + 0.01476             3 -26.385 59.4  0.40  0.229
## 1 -0.5596                            1 -28.841 59.8  0.81  0.186
## 2 -1.1790      +                     2 -27.823 59.9  0.97  0.172
## 8 -5.2870      + 0.03292          +  4 -25.714 60.5  1.49  0.133
## Models ranked by AICc(x)
```

```
write.csv(dd %>% mutate(across(where(is.numeric), ~signif(., 3))),
          "./output/surivalYear1__dredge__model_selection.csv", na = "")

avgmod.2delta <- model.avg(dd, subset = delta < 2, fit = TRUE)
summary(avgmod.2delta)
```

```
##
## Call:
## model.avg(object = get.models(object = dd, subset = delta < 2))
##
## Component model call:
## glm(formula = surviveYear1 ~ <5 unique rhs>, family = binomial, data =
##      dat)
##
## Component models:
##          df logLik  AICc delta weight
## 2         2 -27.34 58.97  0.00   0.28
## 12        3 -26.39 59.37  0.40   0.23
## (Null)    1 -28.84 59.78  0.81   0.19
## 1         2 -27.82 59.94  0.97   0.17
## 123       4 -25.71 60.45  1.49   0.13
##
## Term codes:
##           is_ESE        weanmass is_ESE:weanmass
##                1               2               3
##
## Model-averaged coefficients:
## (full average)
##                       Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)          -2.350338   1.948100    1.975477   1.190    0.234
## weanmass              0.011929   0.014331    0.014528   0.821    0.412
## is_ESETRUE            0.917792   1.740370    1.764684   0.520    0.603
## is_ESETRUE:weanmass -0.003233   0.011544    0.011721   0.276    0.783
##
## (conditional average)
##                      Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)          -2.35034    1.94810     1.97548   1.190    0.234
## weanmass              0.01860    0.01401     0.01432   1.299    0.194
## is_ESETRUE            1.71979    2.07277     2.11094   0.815    0.415
## is_ESETRUE:weanmass -0.02429    0.02213     0.02282   1.065    0.287
```

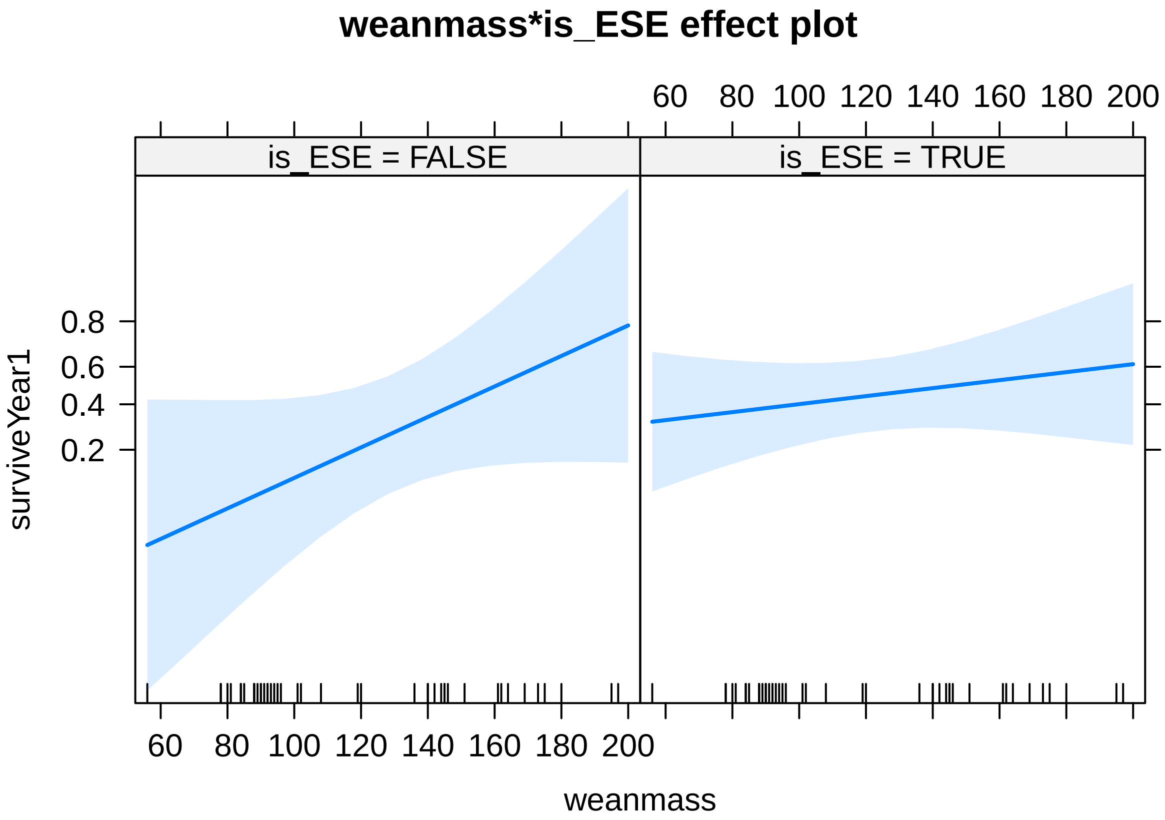
```
importance(avgmod.2delta)
```

```
##                      weanmass is_ESE is_ESE:weanmass
## Sum of weights:      0.64     0.53   0.13
## N containing models:    3        3      1
```

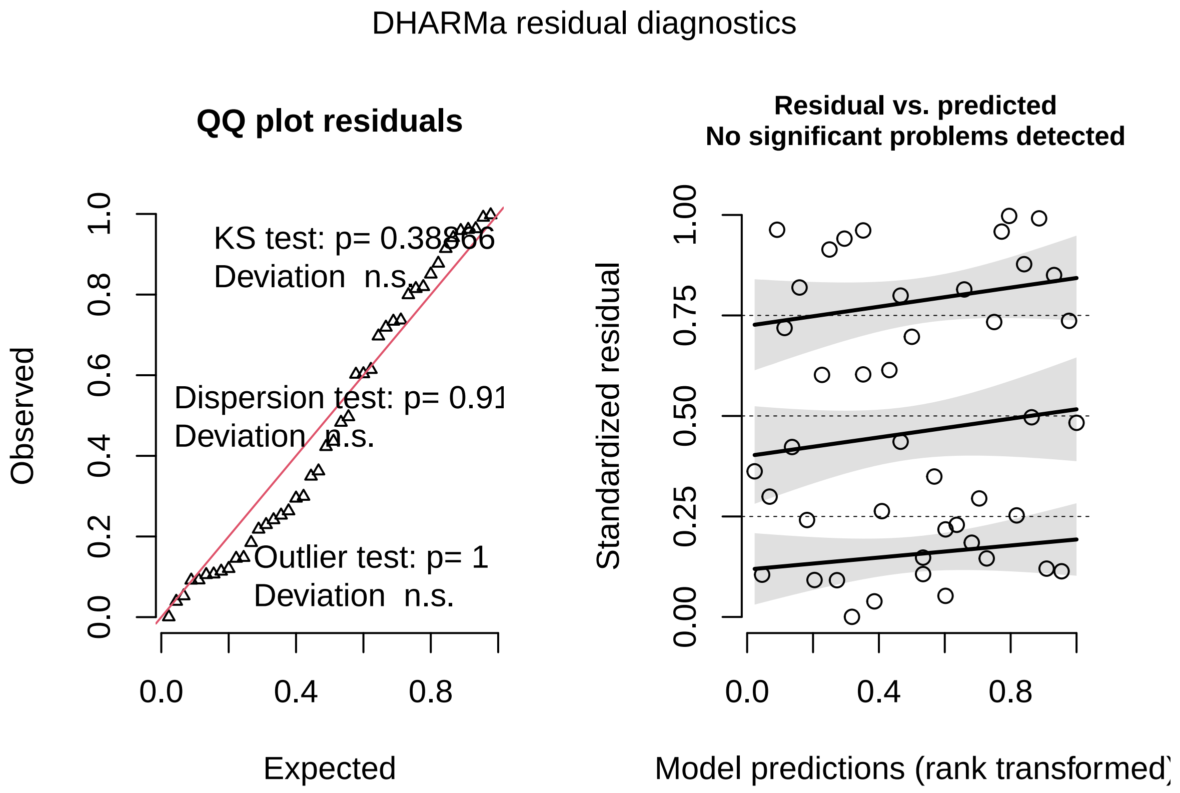
```
# write.csv(summary(avgmod.2delta)$coefmat.subset %>% data.frame(), "./output/1st year survival model t

# Test assumptions
m.final <- glm(surviveYear1 ~  weanmass*is_ESE, data = dat, family = binomial)
summary(m.final)
```

```
##
## Call:
## glm(formula = surviveYear1 ~ weanmass * is_ESE, family = binomial,
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3670  -0.9784  -0.4674   1.1159   2.2260
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -5.28728    2.64837  -1.996   0.0459 *
## weanmass             0.03292    0.01967   1.674   0.0942 .
## is_ESETRUE           4.02276    2.94581   1.366   0.1721
## weanmass:is_ESETRUE -0.02429    0.02213  -1.098   0.2723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 57.682  on 43  degrees of freedom
## Residual deviance: 51.427  on 40  degrees of freedom
## AIC: 59.427
##
## Number of Fisher Scoring iterations: 4
```

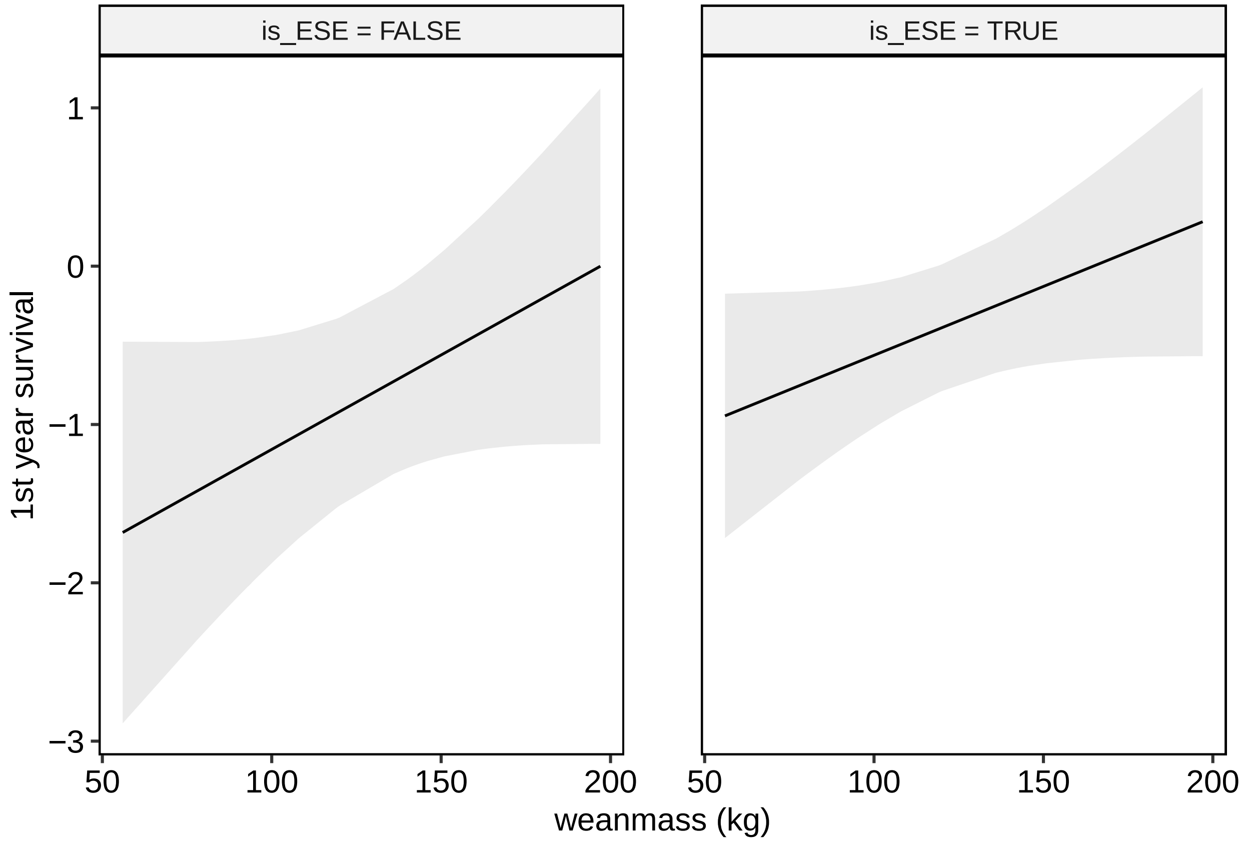
```
effects::allEffects(m.final) %>% plot()
```



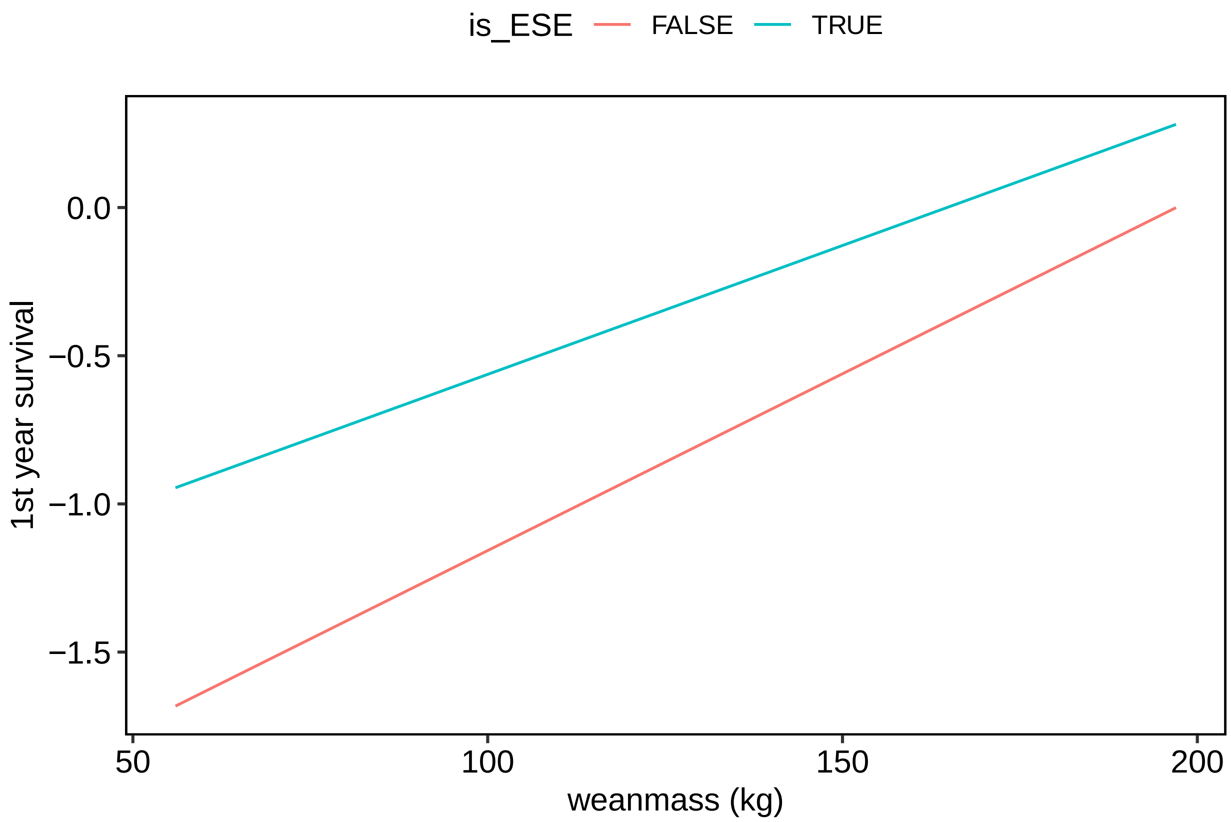
```
simulationOutput <- simulateResiduals(fittedModel = m.global, n = 1000)
plot(simulationOutput)
```



```
# ~ plot avg model ---------------------------------------------------------
a <- tibble(is_ESE = TRUE, weanmass = dat$weanmass)
b <- tibble(is_ESE = FALSE, weanmass = dat$weanmass)

pred <- predict(avgmod.2delta, newdata = a, se.fit = TRUE) %>% as_tibble() %>%
  bind_cols(tibble(is_ESE = TRUE, weanmass = dat$weanmass))

pred <- pred %>% bind_rows(
  predict(avgmod.2delta, newdata = b, se.fit = TRUE) %>% as_tibble() %>% bind_cols(tibble(is_ESE = FALS
)

# config facet labels
withFlow_labs <- c("is_ESE = FALSE", "is_ESE = TRUE")
names(withFlow_labs) <- c("FALSE", "TRUE")

# png('./output/survival model/1st year survival -- avg model effects.png', width=20, height=13, units=
  ggplot(data = pred, aes(x = weanmass)) +
    geom_line(aes(y = fit)) +
    # geom_point(aes(y = fit)) +
    geom_ribbon(aes(ymin = fit-se.fit, ymax = fit+se.fit), alpha = 0.1) +
    facet_wrap(~is_ESE, labeller = labeller(is_ESE = withFlow_labs)) +
    labs(y = "1st year survival", x = "weanmass (kg)") +
    theme_pubr(border = T) +
    theme(panel.spacing = unit(2, "lines"))
```



```
# dev.off()

ggplot(data = pred, aes(x = weanmass, group = is_ESE, colour = is_ESE)) +
  geom_line(aes(y = fit)) +
  # geom_line(aes(y = fit - se.fit), linetype = "dashed") +
  # geom_line(aes(y = fit + se.fit), linetype = "dashed") +
  labs(y = "1st year survival", x = "weanmass (kg)") +
  theme_pubr(border = T)
```



```
#
# # 1st trip to 1st year survival -------------------------------------------
# # create post trip survival dataframe
# sur <- dr %>%
#   group_by(id) %>%
#   summarise(trip_survive = ifelse(first(seen_6m) == TRUE | first(is_trip_complete == TRUE), 1, 0),
#             seen_1y = first(seen_1y),
#             weanmass = first(weanmass),
#             ew_zone = first(ew_zone),
#             birthlocation = first(birthlocation),
#             fieldseason = first(fieldseason)) %>%
#   mutate(post_trip_died = trip_survive - seen_1y,
#          post_trip_died = as.logical(post_trip_died),)
#
# sur_trip1 <- sur %>%
#   filter(trip_survive == 1)
#
# t.test(sur_trip1$weanmass[sur_trip1$post_trip_died == TRUE], sur_trip1$weanmass[sur_trip1$post_trip_d
#
#
# sur_trip1 %>%
#   ggplot(aes(y = weanmass, x = post_trip_died)) +
#   geom_boxplot()
#
# ## Lighter seals died after their first trip.
#
```

```
#
# # Weanmass vs East/West dispersal -----------------------------------------
# sur %>%
#   ggplot(aes(y = weanmass, x = ew_zone)) +
#   geom_boxplot()
#
#
# # Weanmass vs birth location ----------------------------------------------
# sur %>%
#   ggplot(aes(y = weanmass, x = birthlocation)) +
#   geom_boxplot(aes(fill = seen_1y), position=position_dodge(.9))
#
# sur %>%
#   ggplot(aes(x = birthlocation)) +
#   geom_bar(aes(fill = seen_1y)) +
#   facet_wrap(~ew_zone)
#
# sur %>%
#   ggplot(aes(x = fieldseason)) +
#   geom_bar(aes(fill = seen_1y)) +
#   facet_wrap(~ew_zone)
#
# sur %>%
#   ggplot(aes(y = weanmass, x = fieldseason)) +
#   geom_boxplot(aes(fill = seen_1y), position=position_dodge(.9))
#
# sur %>%
#   ggplot(aes(x = birthlocation)) +
#   geom_bar(position = 'stack', aes(fill = fieldseason)) +
#   scale_fill_bluebrown()


# Save Workspace ----------------------------------------------------------


save.image('11_survival_model_workspace.RData')
```