R code PA-CR model

Marwan Naciri

2023-12-13

This code generates the results and the main figures presented in the manuscript "Offspring number, size, and survival: state-dependent optimization of litter size in a long-lived capital breeder" *Ecosphere*

Marwan Naciri, Jon Aars, Magnus Anderson, Marie-Anne Blanchet, Andrew E. Derocher, Marlène Gamelon, Øystein Wiig, and Sarah Cubaynes

A. Run the model

1. Build the model

```
PA_CR <- nimbleCode({
  # ++++++++++++++++++++++++++++++++ priors ++++++++++++++++++++++++++++++++++++
  # Length
  for (i in 1:n_coef_mL) {
    alpha[i] ~ dnorm(0, sd = 5)
  }
  sigma_tau ~ dunif(0, 10)

  # Cohort random effect
  for (i in 1:n_cohort) {
    zeta[i] ~ dnorm(0, sd = sigma_zeta)
  }
  sigma_zeta ~ dunif(0, 10)

  # Litter size
  for (i in 1:n_coef_LS) {
    beta[i] ~ dnorm(0, sd = 1.5)
  }

  # Girth
  for (i in 1:n_coef_mG) {
    gamma[i] ~ dnorm(0, sd = 5)
  }
  sigma_xi ~ dunif(0, 10)

  # Cub mass
  for (i in 1:n_coef_M) {
    delta[i] ~ dnorm(0, sd = 5)
  }
  sigma_epsilon ~ dunif(0, 10)

  # Yearly random effects
```

```
for (i in 1:n_year) {
  eta[i] ~ dnorm(0, sd = sigma_eta)
  rho[i] ~ dnorm(0, sd = sigma_rho)
  nu[i] ~ dnorm(0, sd = sigma_nu)
}
sigma_eta ~ dunif(0, 10)
sigma_rho ~ dunif(0, 10)
sigma_nu ~ dunif(0, 10)


# Recapture
for (i in 1:n_coef_p) {
  lambda[i] ~ dnorm(0, sd = 1.5)
}

# Survival
for (i in 1:n_coef_phi) {
  theta[i] ~ dnorm(0, sd = 1.5)
}

# Yearly random effects on recapture
for (t in 1:(n_year-1)) {
  omega[t] ~ dnorm(0, sd = sigma_omega)
}
sigma_omega ~ dunif(0, 10)

# +++++++++++++++++++++++++++ model & likelihood +++++++++++++++++++++++++++++

# length, girth and litter size
for (k in 1:n_litters) {
  # Length
  mu_mL[k] <- alpha[1] * mAgeY[row_litter[k], col_litter[k]] +
    alpha[2] * mAgeM[row_litter[k], col_litter[k]] +
    alpha[3] * mAgeO[row_litter[k], col_litter[k]] +
    alpha[4] * mSpUnk[row_litter[k]] +
    alpha[5] * mSpPel[row_litter[k]] +
    zeta[cohort[k]]
  mL[row_litter[k], col_litter[k]] ~ dnorm(mu_mL[k], sd = sigma_tau)

  # Litter size
  log(q[k, 1]) <- 0

  log(q[k, 2]) <- beta[1] * mAgeY[row_litter[k], col_litter[k]] +
    beta[2] * mAgeM[row_litter[k], col_litter[k]] +
    beta[3] * mAgeO[row_litter[k], col_litter[k]] +
    beta[4] * mL[row_litter[k], col_litter[k]] +
    beta[5] * mL[row_litter[k], col_litter[k]]^2 +
    beta[6] * mAgeY[row_litter[k], col_litter[k]] * DOY[row_litter[k], col_litter[k]] +
    beta[7] * mSpUnk[row_litter[k]] +
    beta[8] * mSpPel[row_litter[k]] +
    eta[col_litter[k]]

  log(q[k, 3]) <- beta[9] * mAgeY[row_litter[k], col_litter[k]] +
```

```
    beta[10] * mAgeM[row_litter[k], col_litter[k]] +
    beta[11] * mAgeO[row_litter[k], col_litter[k]] +
    beta[12] * mL[row_litter[k], col_litter[k]]

  LS[row_litter[k], col_litter[k]] ~ dcat(s[k, 1:3])
  for (i in 1:3) {
    s[k, i] <- q[k, i]/sum(q[k, 1:3])
  }

  LS1[row_litter[k], col_litter[k]] <- 0.5 * (LS[row_litter[k], col_litter[k]] - 2)*(LS[row_litter[k]
  LS2[row_litter[k], col_litter[k]] <- -1 * (LS[row_litter[k], col_litter[k]] - 1)*(LS[row_litter[k],
  LS3[row_litter[k], col_litter[k]] <- 0.5 * (LS[row_litter[k], col_litter[k]] - 1)*(LS[row_litter[k]

  # Girth
  mu_mG[k] <- gamma[1] * mAgeY[row_litter[k], col_litter[k]] +
    gamma[2] * mAgeM[row_litter[k], col_litter[k]] +
    gamma[3] * mAgeO[row_litter[k], col_litter[k]] +
    gamma[4] * mL[row_litter[k], col_litter[k]] +
    gamma[5] * DOY[row_litter[k], col_litter[k]] +
    gamma[6] * (-1) * (LS[row_litter[k], col_litter[k]] - 1)*(LS[row_litter[k], col_litter[k]] - 3) +
    gamma[7] * 0.5 * (LS[row_litter[k], col_litter[k]] - 1)*(LS[row_litter[k], col_litter[k]] - 2) +
    gamma[8] * mSpUnk[row_litter[k]] +
    gamma[9] * mSpPel[row_litter[k]] +
    rho[col_litter[k]]
  mG[row_litter[k], col_litter[k]] ~ dnorm(mu_mG[k], sd = sigma_xi)
}

# Cub mass
for (k in 1:n_cubs) {

  mu_M[k] <- delta[1] * LS1[mother[k], col_cub[k]] * mAgeY[mother[k], col_cub[k]] +
    delta[2] * LS1[mother[k], col_cub[k]] * mAgeM[mother[k], col_cub[k]] +
    delta[3] * LS1[mother[k], col_cub[k]] * mAgeO[mother[k], col_cub[k]] +

    delta[4] * LS1[mother[k], col_cub[k]] * mL[mother[k], col_cub[k]] +
    delta[5] * LS1[mother[k], col_cub[k]] * mG[mother[k], col_cub[k]] +
    delta[6] * LS1[mother[k], col_cub[k]] * DOY[mother[k], col_cub[k]] +
    delta[7] * LS1[mother[k], col_cub[k]] * Sex[k] +
    delta[8] * LS1[mother[k], col_cub[k]] * mSpUnk[mother[k]] +
    delta[9] * LS1[mother[k], col_cub[k]] * mSpPel[mother[k]] +

    delta[10] * LS2[mother[k], col_cub[k]] * mAgeY[mother[k], col_cub[k]] +
    delta[11] * LS2[mother[k], col_cub[k]] * mAgeM[mother[k], col_cub[k]] +
    delta[12] * LS2[mother[k], col_cub[k]] * mAgeO[mother[k], col_cub[k]] +

    delta[13] * LS3[mother[k], col_cub[k]] * mAgeY[mother[k], col_cub[k]] +
    delta[14] * LS3[mother[k], col_cub[k]] * mAgeM[mother[k], col_cub[k]] +
    delta[15] * LS3[mother[k], col_cub[k]] * mAgeO[mother[k], col_cub[k]] +

    delta[16] * (LS2[mother[k], col_cub[k]] + LS3[mother[k], col_cub[k]]) * mL[mother[k], col_cub[k]]
    delta[17] * (LS2[mother[k], col_cub[k]] + LS3[mother[k], col_cub[k]]) * mG[mother[k], col_cub[k]]
    delta[18] * (LS2[mother[k], col_cub[k]] + LS3[mother[k], col_cub[k]]) * DOY[mother[k], col_cub[k]
    delta[19] * (LS2[mother[k], col_cub[k]] + LS3[mother[k], col_cub[k]]) * Sex[k] +
```

```
    delta[20] * (LS2[mother[k], col_cub[k]] + LS3[mother[k], col_cub[k]]) * mSpUnk[mother[k]] +
    delta[21] * (LS2[mother[k], col_cub[k]] + LS3[mother[k], col_cub[k]]) * mSpPel[mother[k]] +

    nu[col_cub[k]]

  M[k] ~ dnorm(mu_M[k], sd = sigma_epsilon)
}

for (i in 1:n_ind) {
  for (t in f[i]:(n_year-1)) {
    # Survival
    logit(phi[i, t]) <-
      theta[1] * Age0[i, t] * mAgeY[mother[i], t] +
      theta[2] * Age0[i, t] * mAgeM[mother[i], t] +
      theta[3] * Age0[i, t] * mAgeO[mother[i], t] +

      theta[4] * Age0[i, t] * M[i] +
      theta[5] * Age0[i, t] * (-1 * (LS[mother[i], t] - 1) * (LS[mother[i], t] - 3) +  # LS2
      0.5 * (LS[mother[i], t] - 1) * (LS[mother[i], t] - 2)) +  # LS3
      theta[6] * Age0[i, t] * mL[mother[i], t] +
      theta[7] * Age0[i, t] * mG[mother[i], t] +
      theta[8] * Age0[i, t] * DOY[mother[i], t] +
      theta[9] * Age0[i, t] * Sex[i] +
      theta[10] * Age0[i, t] * mSpUnk[mother[i]] +
      theta[11] * Age0[i, t] * mSpPel[mother[i]] +

      theta[12] * Age1[i, t] +
      theta[13] * (Age2[i, t] + Age3_4[i, t]) +
      theta[14] * Age5_19[i, t] +
      theta[15] * Age20[i, t] +

      theta[16] * (1 - Age0[i, t]) * Sex[i] +                              # Effect of sex = male f
      theta[17] * ((1 - Age0[i, t]) * (1 - Sex[i]) * mSpUnk[i] +          # Effect of space use = NA
                  Age1[i, t] * Sex[i] * mSpUnk[mother[i]]) +               # Effect of maternal spac
      theta[18] * ((1 - Age0[i, t]) * (1 - Sex[i]) * mSpPel[i] +     # Effect of space use = pelagic
                  Age1[i, t] * Sex[i] * mSpPel[mother[i]])            # Effect of maternal space use

    # Recapture
    logit(p[i, t]) <- lambda[1] +
      lambda[2] * (mSpUnk[i] * (1 - Sex[i]) +                                               # all N
                  mSpUnk[mother[i]] * Sex[i] * (Age1[i, t+1] + Age2[i, t+1])) + # all male depende
      lambda[3] * (mSpPel[i] * (1 - Sex[i]) +                                               # all p
                  mSpPel[mother[i]] * Sex[i] * (Age1[i, t+1] + Age2[i, t+1])) + # all male depende
      lambda[4] * Sex[i] * (Age3_4[i, t+1] + Age5_19[i, t+1] + Age20[i, t+1]) +
      omega[t]
  }
}

for (i in 1:n_ind) {

  # latent state at the first capture
  z[i, f[i]] <- 1
```

```
    for (t in (f[i]+1):n_year) {
      # State process
      z[i, t] ~ dbern(phi[i, t-1] * z[i, t-1])

      # Observation process
      y[i, t] ~ dbern(p[i, t-1] * z[i, t])

    } # t
  } # i
})
```

2. Process the data

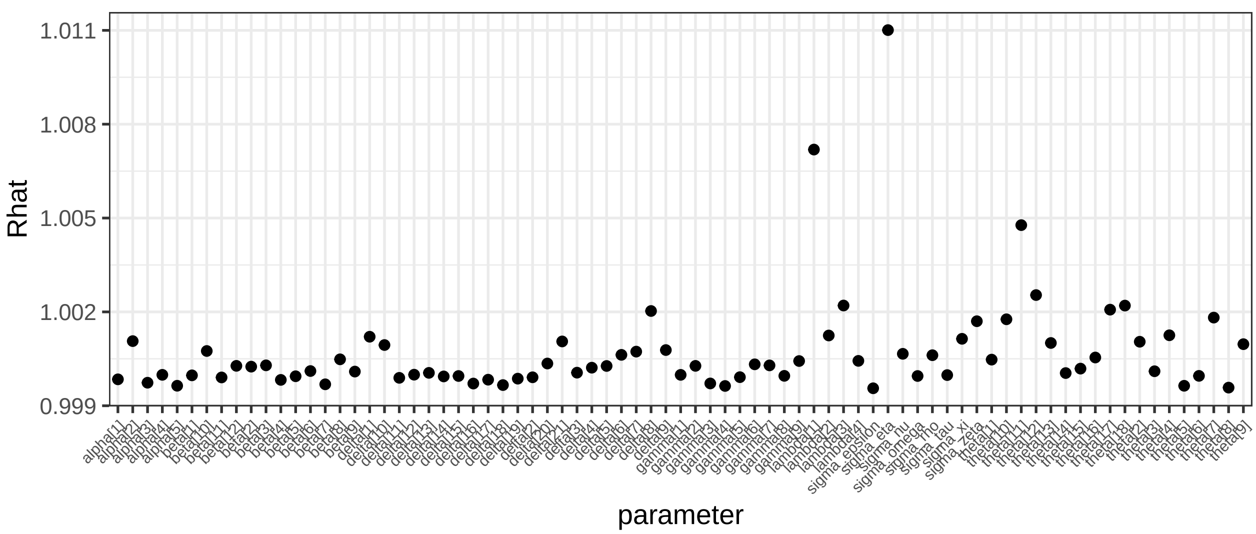
3. Run the model

B. Inspect the results

1. Diagnostic plots

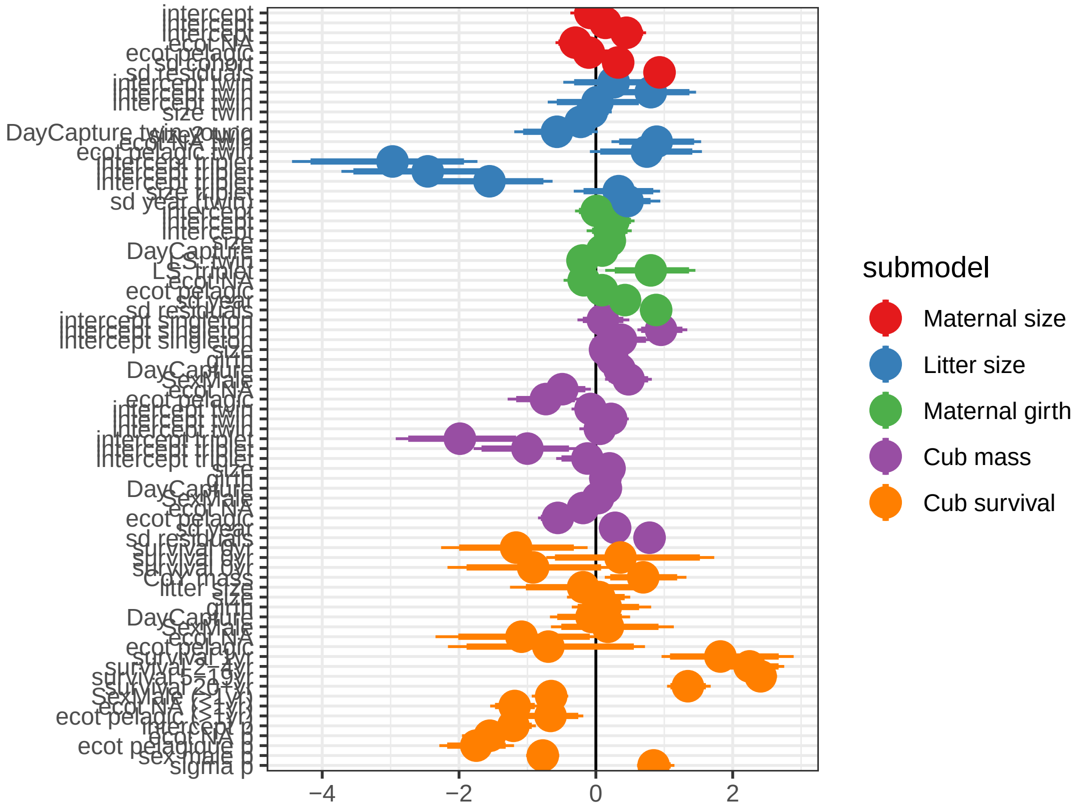
2. Rhat

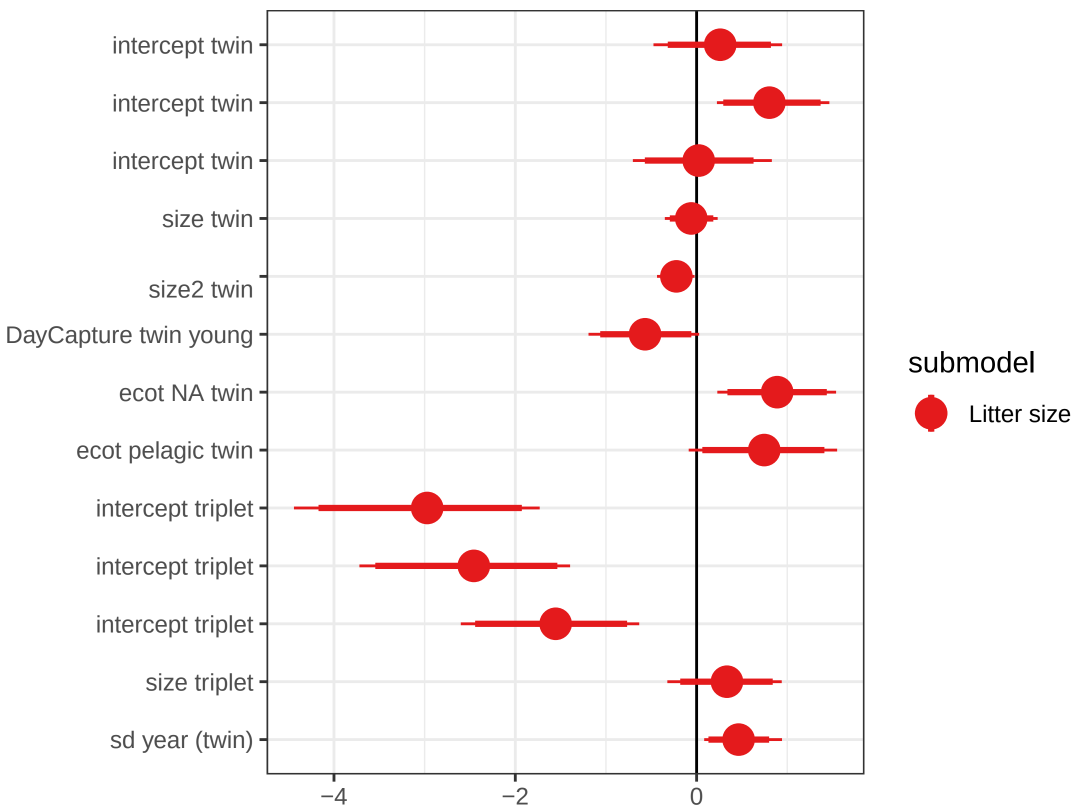
Calculate the Rhat value of each parameter, to make sure the chains have converged (Rhat< 1.1)

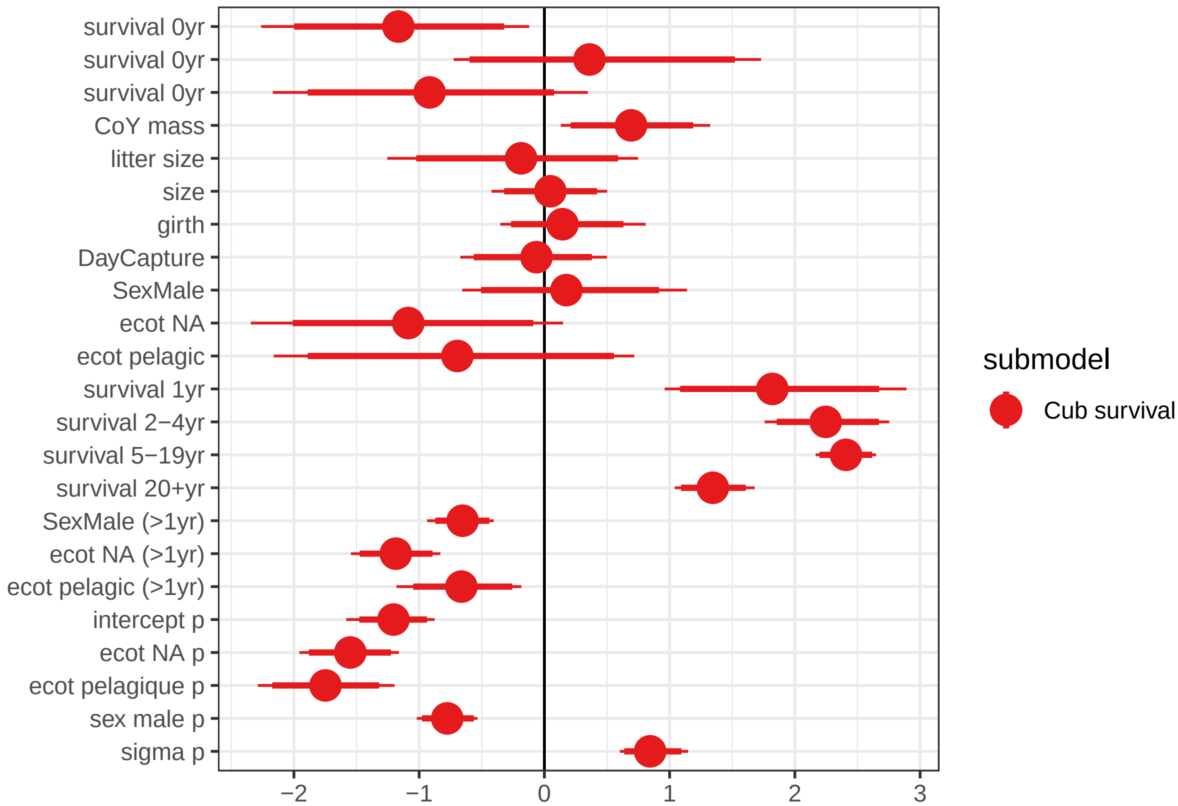


All Rhat are < 1.1

3. Caterpillar plots







C. Compute residuals

The residuals computed here will be used to plot the observations corrected for variables other than that on the x axis for the figures.

a. Cub mass for Fig. 2A

b. Cub mass for Fig. 2B

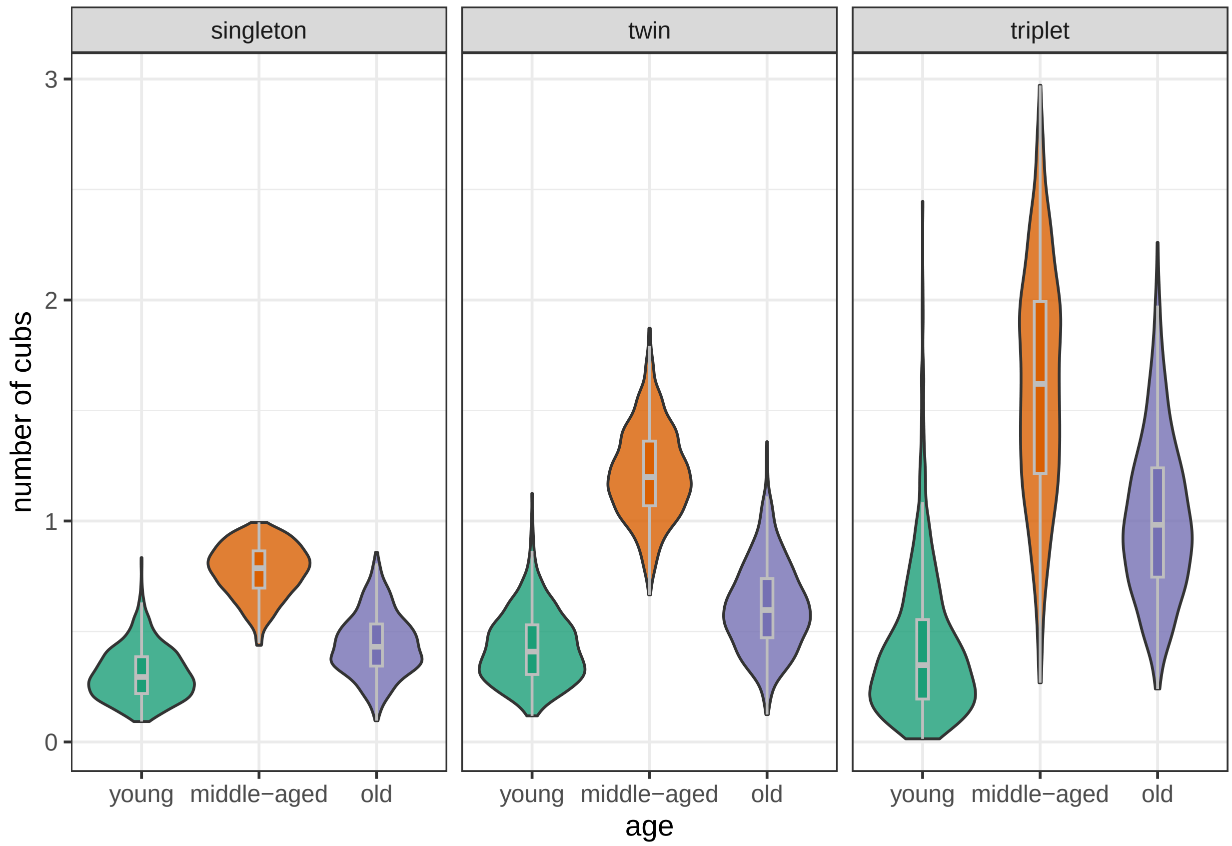
c. Cub mass for Fig. 2C

D. Predict number of surviving cubs

Let's predict the number of surviving cubs by litter size and maternal age

We need to account for the fact that females who produce singleton/twin/triplet litters are not the same size on average. Let's compute the size of females corrected for the cohort random effect and the effect of the space-use strategy:

```
## `summarise()` has grouped output by 'iteration'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'iteration'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'iteration'. You can override using the
## `.groups` argument.
```



E. Compute probabilities of direction (pd)

1. Pd and effect sizes

a. Maternal length

```
## [1] 178 213
```

```
## [1] 1.37871
```

```
##       2.5%      97.5%
## -0.4264643  3.1591493
```

```
## [1] 0.933
```

```
## [1] 1.960729
```

```
##       2.5%      97.5%
## -0.1311223  4.0995814
```

```
## [1] 0.966
```

```
## [1] 3.339439
```

```
##     2.5%    97.5%
## 1.121562 5.605809
```

```
## [1] 0.9985
```

```
## [1] -0.2983274
```

```
##        2.5%       97.5%
```

```
## -0.57634196 -0.02061026
```

```
## [1] 0.98125
```

```
## [1] -0.1042238
```

```
##       2.5%      97.5%
## -0.4529151  0.2664911
```

```
## [1] 0.70825
```

```
## [1] 0.1133352
```

```
##       2.5%      97.5%
## 0.02402668 0.27528545
```

b. Litter size

```
## [1] 1.688
```

```
## [1] 0.5289377
```

```
## [1] 0.1224279
```

```
##        2.5%       97.5%
## -0.03107809  0.27701748
```

```
## [1] 0.93875
```

```
## [1] 0.2090952
```

```
##       2.5%      97.5%
## 0.03307916 0.38189636
```

```
## [1] 0.99025
```

```
## [1] 0.08666729
```

```
##       2.5%      97.5%
## -0.1038202  0.2738608
```

```
## [1] 0.99025
```

```
## [1] -0.2226712
```

```
##        2.5%       97.5%
## -0.44503442 -0.01327327
```

```
## [1] 0.983
```

```
## [1] 0.97075
```

```
## [1] 0.9965
```

```
## [1] 0.96
```

```
## [1] 0.2593803
```

```
##        2.5%       97.5%
## 0.007015413 0.832116116
```

c. Maternal girth

```
## [1]  94 135
```

```
## [1] 1.777517
```

```
##        2.5%       97.5%
## -0.08527602  3.73335906
```

```
## [1] 0.9675
```

```
## [1] 0.4558738
```

```
##      2.5%     97.5%
## -1.855995  2.802063
```

```
## [1] 0.648
```

```
## [1] 1.321643
```

```
##      2.5%     97.5%
## -1.065840  3.786976
```

```
## [1] 0.8545
```

```
## [1] 1.436735
```

```
##       2.5%      97.5%
## -0.4406985  3.1938666
```

```
## [1] 0.93675
```

```
## [1] -5.784318
```

```
##        2.5%       97.5%
## -10.7086933  -0.9563209
```

```
## [1] 0.9885
```

```
## [1] 0.999
```

```
## [1] 0.907
```

```
## [1] 0.87075
```

```
## [1] 0.66075
```

```
## [1] 0.1920406
```

```
##       2.5%      97.5%
## 0.06021129 0.40962968
```

d. Cub mass

```
## [1]  3.00 31.25
```

```
## [1] 1.220548
```

```
##     2.5%    97.5%
## 1.110384 1.339997
```

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

```
## [1] 1.48271
```

```
##     2.5%    97.5%
## 1.284317 1.699458
```

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

```
## [1] 1.215447
```

```
##     2.5%    97.5%
## 1.070512 1.376433
```

```
## [1] 0.99975

## [1] 1.309397

##     2.5%    97.5%
## 1.143869 1.489855

## [1] 1

## [1] 1.201367

##     2.5%    97.5%
## 1.034060 1.388952

## [1] 0.99075

## [1] 1.093468

##      2.5%     97.5%
## 0.9428594 1.2618430

## [1] 0.875

## [1] 1.101892

##     2.5%    97.5%
## 1.024534 1.178438

## [1] 0.998

## [1] 1.054262

##      2.5%     97.5%
## 0.9651679 1.1496110

## [1] 0.878

## [1] 1.046729

##      2.5%     97.5%
## 0.9536279 1.1460931

## [1] 0.82325

## [1] 0.1975184

##       2.5%      97.5%
## 0.02568826 0.37502017

## [1] 0.989

## [1] 0.1511139

##        2.5%       97.5%
## -0.05238364  0.34621105

## [1] 0.93275

## [1] 0.04640455

##       2.5%      97.5%
## -0.1418984  0.2439453

## [1] 0.6885

## [1] 0.93675

## [1] 1
```

```
## [1] 0.99475
```

```
## [1] 0.99775
```

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

```
## [1] 0.99625
```

```
## [1] 1.166981
```

```
##     2.5%    97.5%
## 1.043683 1.305583
```

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

```
## [1] 0.99025
```

```
## [1] 0.94575
```

```
## [1] 0.99475
```

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

```
## [1] 0.9945
```

```
## [1] 0.63925
```

```
## [1] 0.08466851
```

```
##       2.5%      97.5%
## 0.02281089 0.18855145
```

e. Litter mass

```
## singleton      twin   triplet
##  15.38750  25.23169  31.24473
```

```
## [1] 1.642374
```

```
##     2.5%    97.5%
## 1.492541 1.801179
```

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

```
## [1] 1.239089
```

```
##     2.5%    97.5%
## 1.089774 1.401199
```

```
## [1] 0.99975
```

f. Cub survival

```
## [1] 0.9925
```

```
## [1] 0.3280594
```

```
##      2.5%     97.5%
## 0.1088579 0.5735900
```

```
## [1] 0.995
```

```
## [1] 0.2787698
```

```
##       2.5%      97.5%
## 0.03098407 0.53984559
```

```
## [1] 0.985
```

```
## [1] -0.04928959
```

```
##       2.5%      97.5%
## -0.2751187  0.1624413
```

```
## [1] 0.66625
```

```
## [1] 0.62625
```

```
## [1] -0.1862317
```

```
## [1] 0.5825
```

```
## [1] 0.0472786
```

```
## [1] 0.6875
```

```
## [1] 0.5525
```

```
## [1] 0.64375
```

```
## [1] 0.9575
```

```
## [1] 0.83125
```

g. Recapture probability

```
## [1] 0.2313914
```

```
##      2.5%     97.5%
## 0.1737597 0.2942490
```

```
## [1] 0.06135659
```

```
##       2.5%      97.5%
## 0.03853024 0.09083818
```

```
## [1] 0.05151737
```

```
##       2.5%      97.5%
## 0.02829964 0.08469957
```

```
## [1] 0.1224796
```

```
##       2.5%      97.5%
## 0.08872552 0.16188407
```

```
## [1] 0.8484879
```

```
##      2.5%     97.5%
## 0.6062677 1.1740871
```

h. Productivity by litter size

```
## [1] 0.1174675
```

```
##       2.5%      97.5%
## -0.1926263  0.4087787
```

```
## [1] 0.8125
```

```
## [1] 0.4410225
```

```
##      2.5%     97.5%
## 0.1159444 0.7643456
```

```
## [1] 0.99125
```

```
## [1] 0.1699285
```

```
##       2.5%      97.5%
## -0.1706343  0.5126746
```

```
## [1] 0.8325
```

```
## [1] 0.005329859
```

```
##       2.5%      97.5%
## -0.6805737  0.4609637
```

```
## [1] 0.5775
```

```
## [1] -0.397805
```

```
##       2.5%      97.5%
## -1.1397305  0.4784175
```

```
## [1] 0.8125
```

```
## [1] -0.4021119
```

```
##         2.5%        97.5%
## -0.974152012  0.001475672
```

```
## [1] 0.97375
```

2. Variance explained by model

```
## [1] 0.132007
```

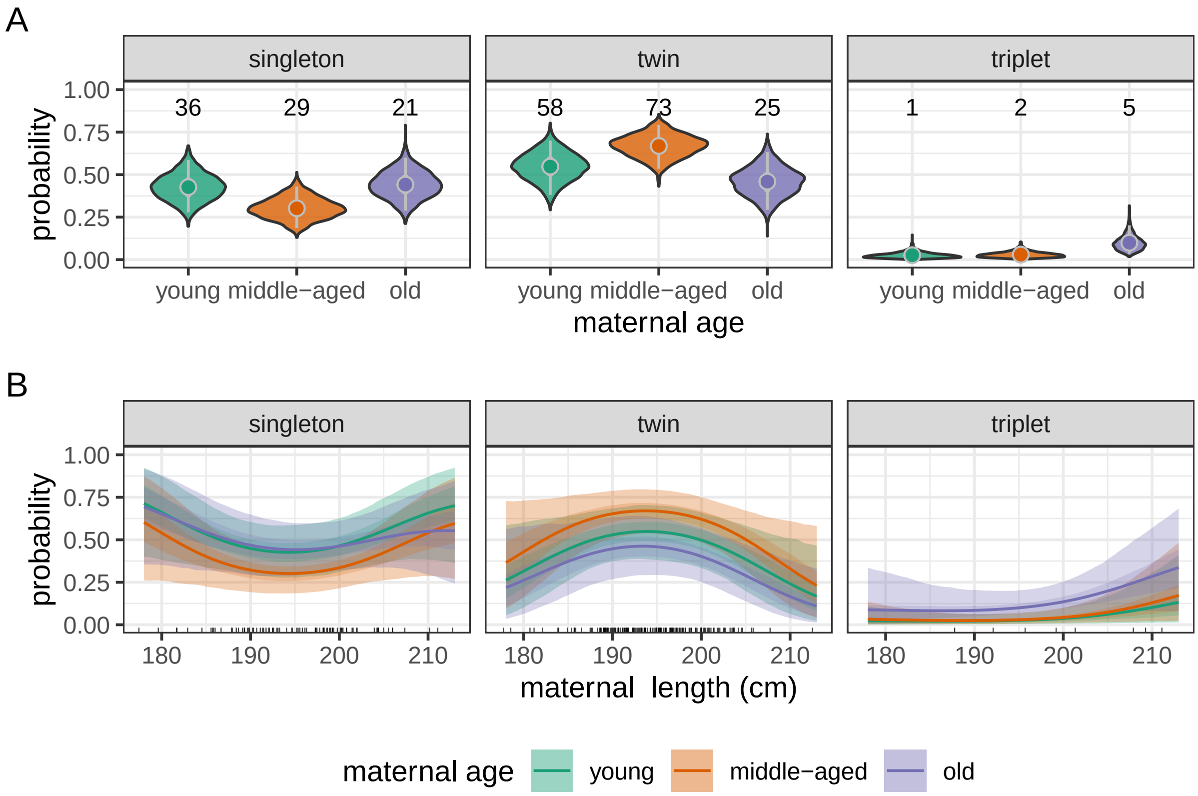
```
## [1] 0.2197837
```

```
## [1] 0.3834601
```

F. Plot main figures

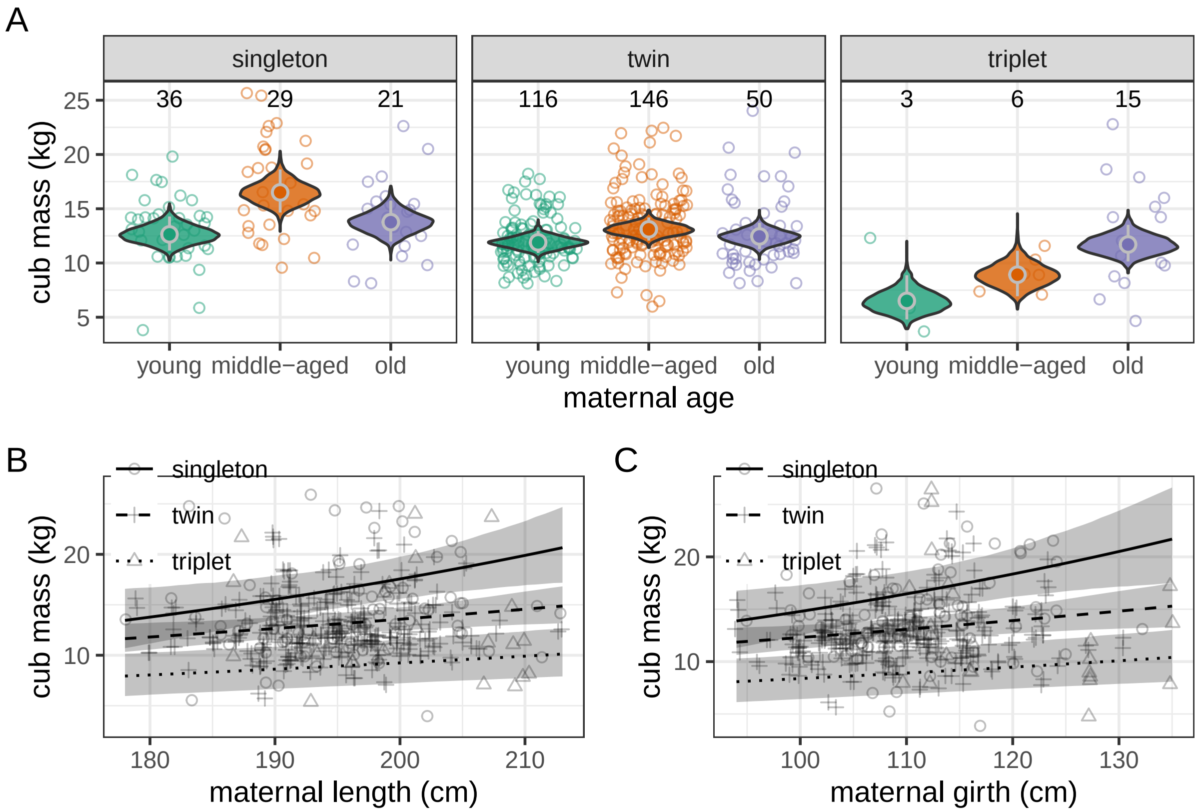
1. Figure 2: Effects on litter size

```
## `summarise()` has grouped output by 'var', 'litter_size'. You can override
## using the `.groups` argument.
```



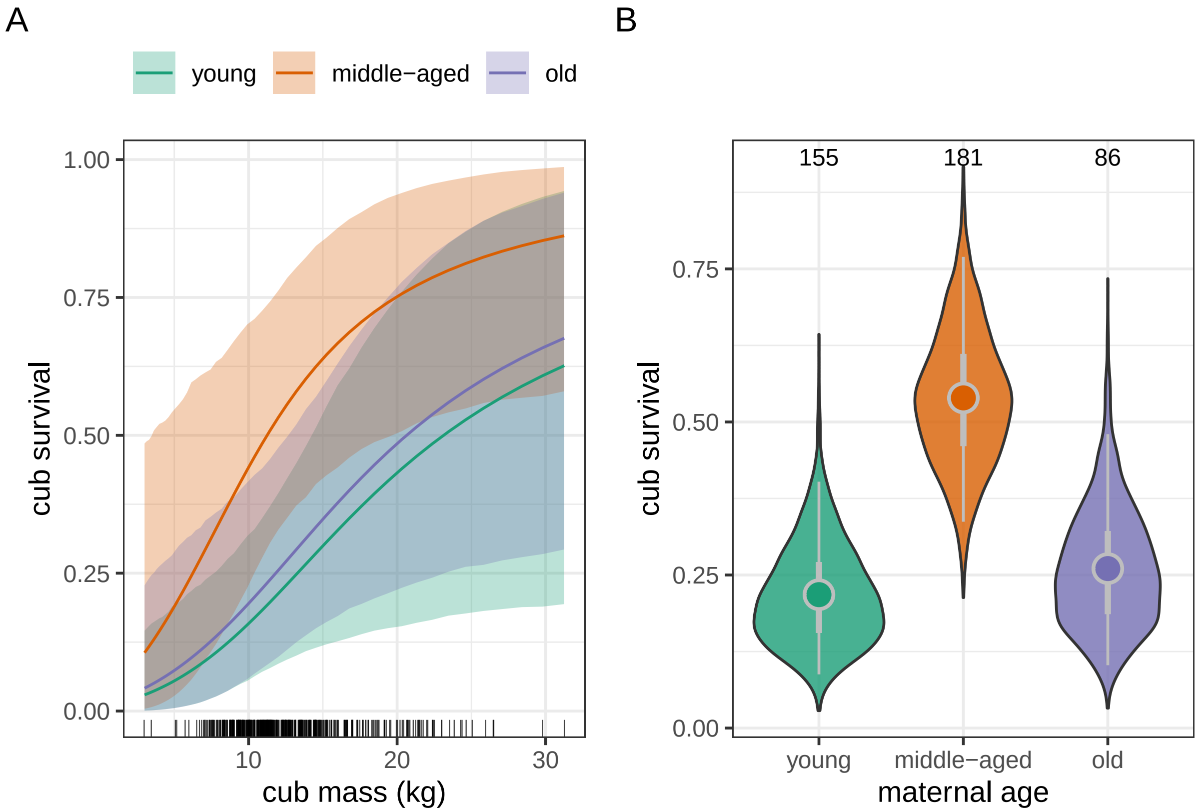
2. Figure 3: Determinants of cub mass

```
## 'summarise()' has grouped output by 'var'. You can override using the '.groups'
## argument.

## Warning: A numeric 'legend.position' argument in 'theme()' was deprecated in ggplot2
## 3.5.0.
## i Please use the 'legend.position.inside' argument of 'theme()' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

## 'summarise()' has grouped output by 'var'. You can override using the '.groups'
## argument.
```



3. Figure 4: Determinants of cub survival

```
## 'summarise()' has grouped output by 'var'. You can override using the '.groups'
## argument.
```


A
B
young
middle−aged
old
1.00
0.75
0.50
0.25
0.00
cub survival
10
20
30
cub mass (kg)
155
181
86
0.75
0.50
0.25
0.00
cub survival
young
middle−aged
old
maternal age

4. Figure 6: Productivity by litter size