R code PA-CR model

Marwan Naciri

2023-12-13

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

Marwan Naciri, Marlène Gamelon, Jon Aars, Marie-Anne Blanchet, 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) {
```

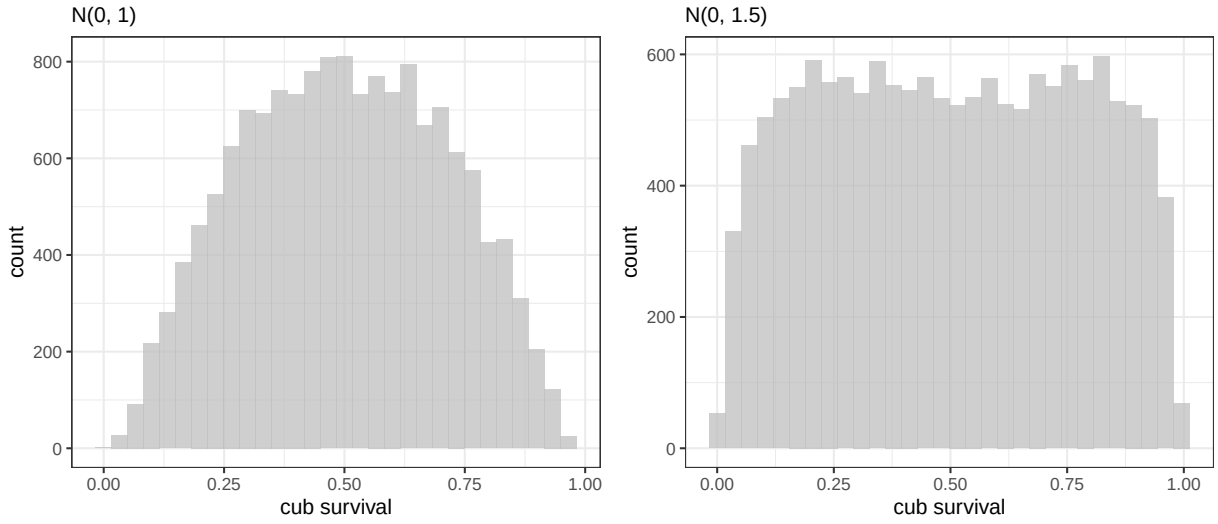
```
  eta[i] ~ dnorm(0, sd = sigma_eta)
  rho[i] ~ dnorm(0, sd = sigma_rho)
  nu[i] ~ dnorm(0, sd = sigma_nu)
}
sigma_eta ~ dunif(0, 10)
sigma_rho ~ dunif(0, 10)
sigma_nu ~ dunif(0, 10)


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

# Survival
theta[1] ~ dnorm(0, sd = 1.5)
theta[2] ~ dnorm(0, sd = 1)
for (i in 3: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])
  }

  # 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] * LS2[row_litter[k], col_litter[k]] +
    gamma[7] * LS3[row_litter[k], col_litter[k]] +
    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] * (LS2[mother[i], t] + LS3[mother[i], t]) +
      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] +
      theta[14] * Age3_4[i, t] +
      theta[15] * Age5_19[i, t] +
      theta[16] * Age20[i, t] +

      theta[17] * (1 - Age0[i, t]) * Sex[i] +                              # Effect of sex = male f
      theta[18] * ((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[19] * ((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] * (Age0[i, t+1] + Age1[i, t+1]+ Age2[i, t+1])) + # al
      lambda[3] * (mSpPel[i] * (1 - Sex[i]) +                                                    # all p
                     mSpPel[mother[i]] * Sex[i] * (Age0[i, t+1] + Age1[i, t+1] + Age2[i, t+1])) + # a
      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
})
```

Here, I use a relatively informative prior for theta[2] (survival of cubs of middle-aged females) because otherwise, I get unrealistically high survival (mean = 0.77). This informative prior is N(0, 1), to compare to the less informative N(0, 1.5):



2. Process the data

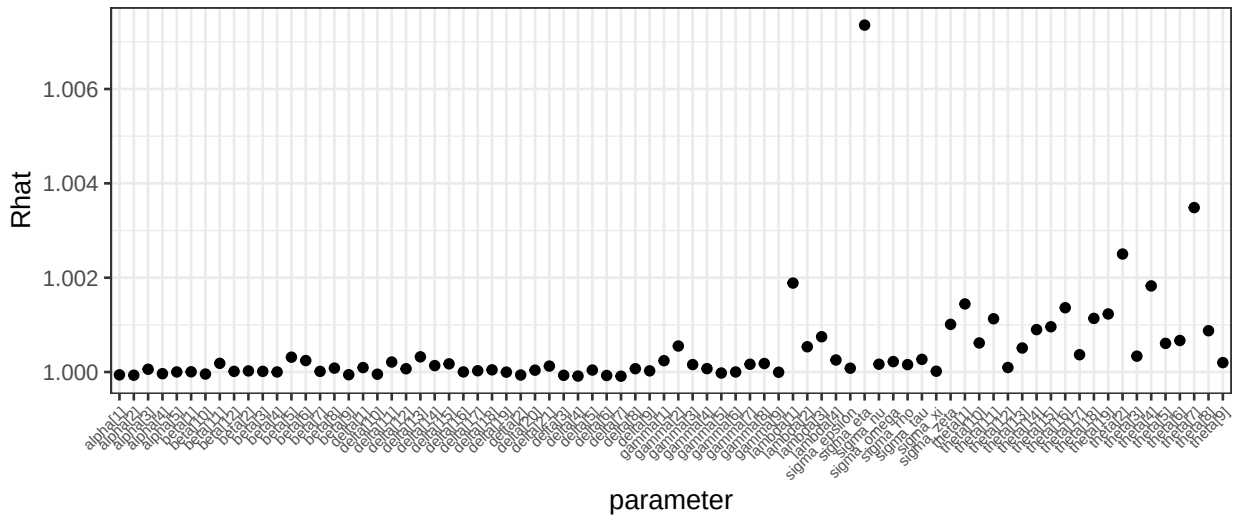
```
## Using litter_size as value column: use value.var to override.
```

B. Inspect the results

1. Trace plots

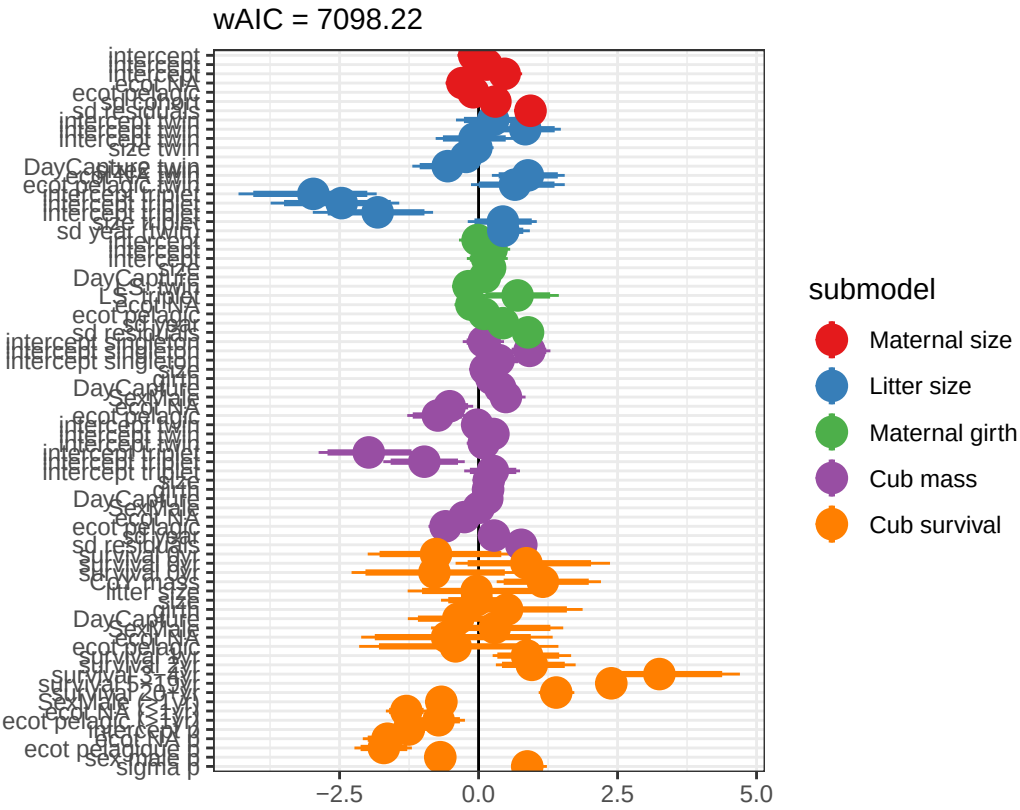
2. Rhat

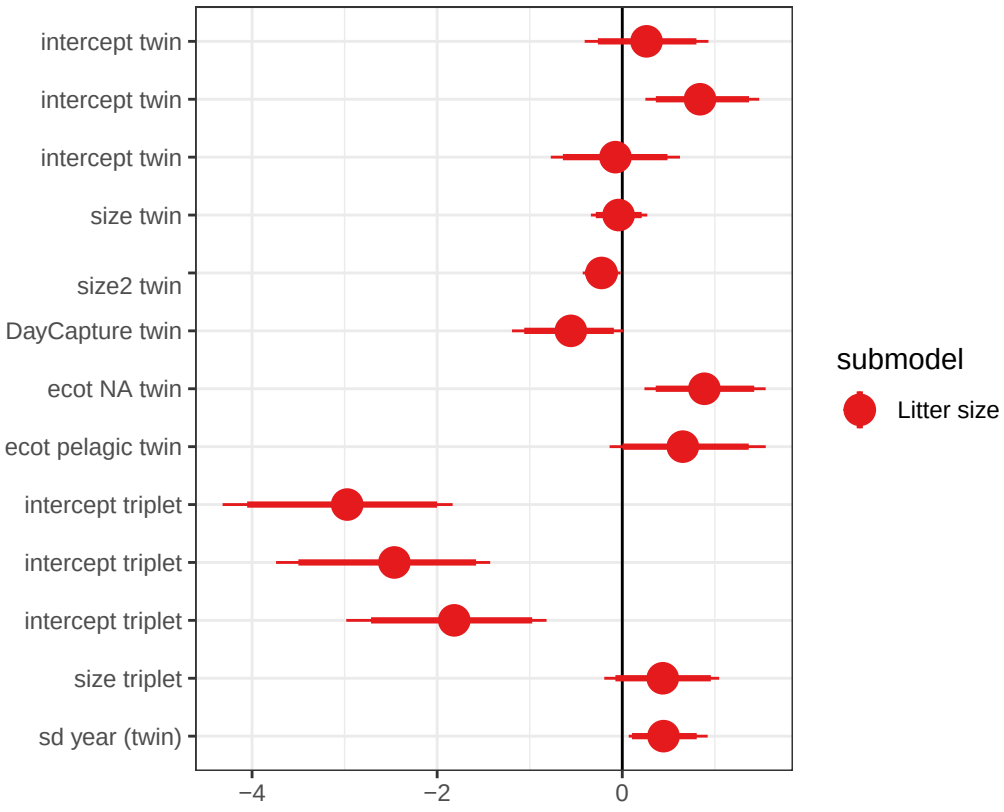
Calculate the Rhat value of each parameter, to make sure the chains have converged (Rhat$<$ 1.1)

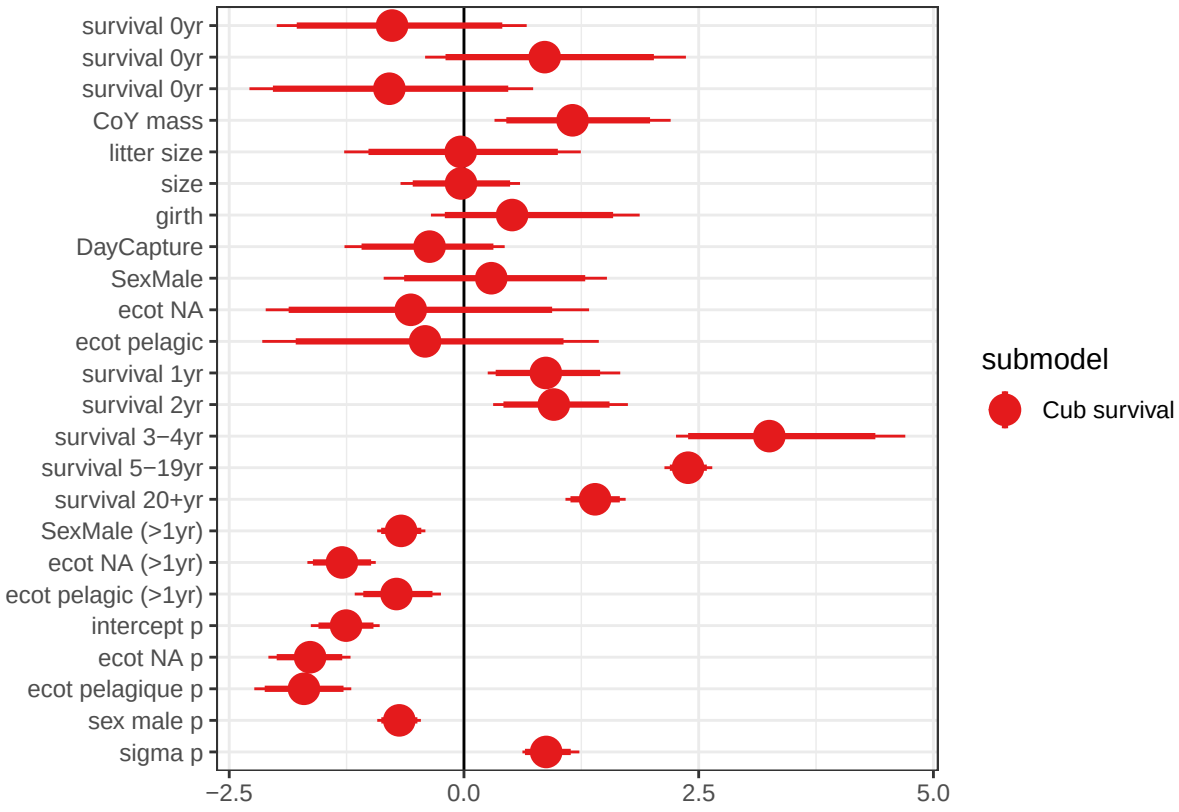


All Rhat are < 0.01

3. Caterpillar plots

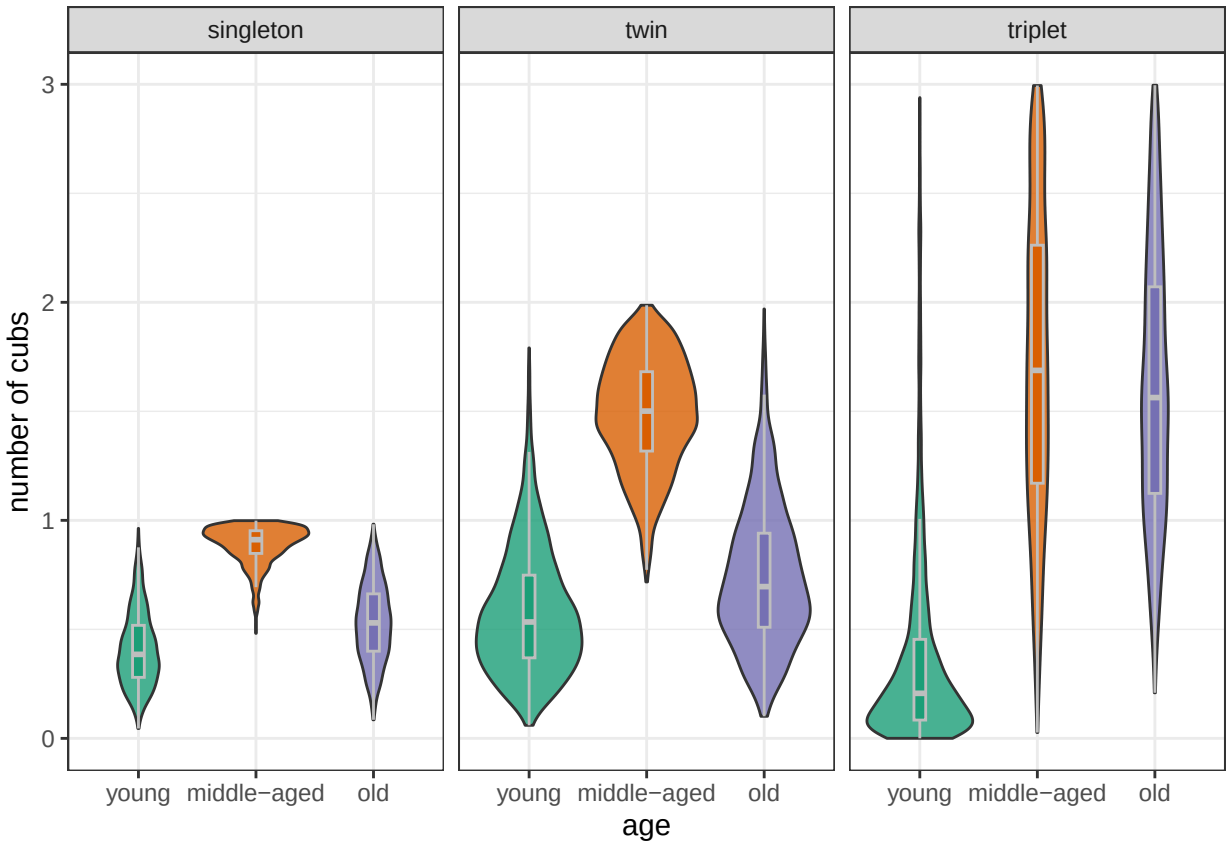



intercept twin
intercept twin
intercept twin
size twin
size2 twin
DayCapture twin
ecot NA twin
ecot pelagic twin
intercept triplet
intercept triplet
intercept triplet
size triplet
sd year (twin)
−4
−2
0
submodel
Litter size


survival 0yr
survival 0yr
survival 0yr
CoY mass
litter size
size
girth
DayCapture
SexMale
ecot NA
ecot pelagic
survival 1yr
survival 2yr
survival 3−4yr
survival 5−19yr
survival 20+yr
SexMale (>1yr)
ecot NA (>1yr)
ecot pelagic (>1yr)
intercept p
ecot NA p
ecot pelagique p
sex male p
sigma p
−2.5
0.0
2.5
5.0
submodel
Cub survival

C. Compute residuals and predictions

1. Predict number of surviving CoYs by litter size and maternal age



2. Compute residuals

The residuals computed here will be used to plot the observations corrected for other variables for the figures.

a. Cub mass

b. Cub mass for plot length

c. Cub mass for plot girth

3. Plot random effects

Here I convert the random effects to the natural scale for continuous variables, and to the probability scale for probabilities.

```
## Warning: Removed 57600 rows containing non-finite values (‘stat_summary()‘).
```

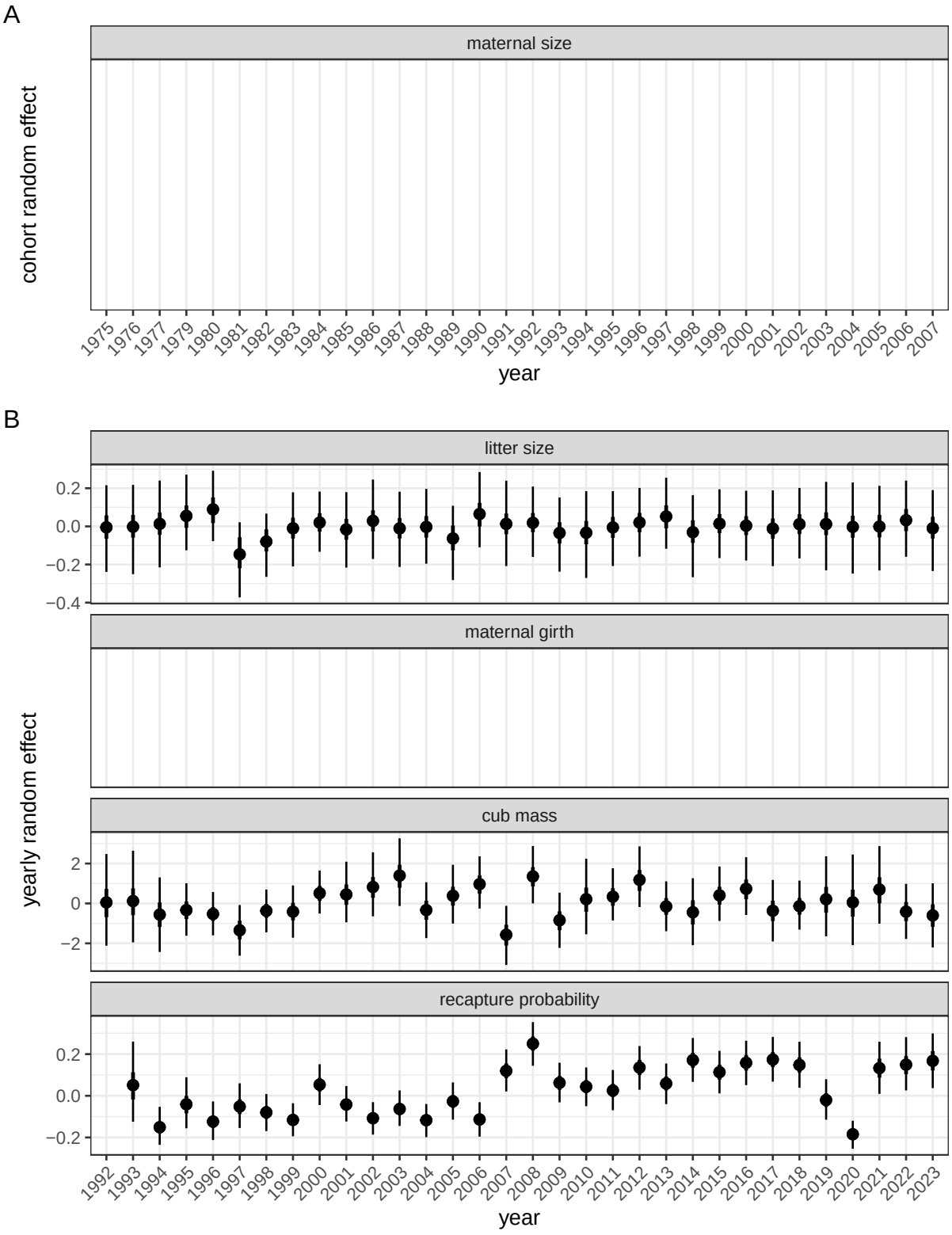
```
## Warning in max(f): aucun argument pour max ; -Inf est renvoyé
```

```
## Warning: Computation failed in ‘stat_summary()‘
## Caused by error in ‘seq_len()‘:
## ! l’argument doit être convertible automatiquement en un entier non négatif
```

```
## Warning: Removed 57600 rows containing non-finite values (‘stat_summary()‘).
```

```
## Warning in max(f): aucun argument pour max ; -Inf est renvoyé
```

```
## Warning: Computation failed in ‘stat_summary()‘
## Caused by error in ‘seq_len()‘:
## ! l’argument doit être convertible automatiquement en un entier non négatif
## Warning: Removed 59400 rows containing non-finite values (‘stat_summary()‘).
## Removed 59400 rows containing non-finite values (‘stat_summary()‘).
```


A
maternal size
cohort random effect
1975
1976
1977
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
year
B
litter size
0.2
0.0
−0.2
−0.4
maternal girth
cub mass
2
0
−2
recapture probability
0.2
0.0
−0.2
yearly random effect
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
year

D. Compute pd and other quantities

1. Pd and effect sizes

a. Maternal length

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

```
## [1] 1.40679
```

```
##       2.5%      97.5%
## -0.4167219  3.2173274
```

```
## [1] 0.9372778
```

```
## [1] 2.106713
```

```
##       2.5%      97.5%
## -0.1090095  4.3281067
```

```
## [1] 0.9692778
```

```
## [1] 3.513503
```

```
##     2.5%    97.5%
## 1.180246 5.859049
```

```
## [1] 0.9984444
```

```
## [1] -0.2922503
```

```
##         2.5%        97.5%
## -0.576971752 -0.006194798
```

```
## [1] 0.9774444
```

```
## [1] -0.09573022
```

```
##       2.5%      97.5%
## -0.4653858  0.2794673
```

```
## [1] 0.6937222
```

```
## [1] 0.1034846
```

```
##        2.5%       97.5%
## 0.003769299 0.266616013
```

b. Litter size

```
## [1] 1.683128
```

```
## [1] 0.5246046
```

```
## [1] 0.1238265
```

```
##        2.5%       97.5%
## -0.02789276  0.28134005
```

```
## [1] 0.9426667
```

```
## [1] 0.2311631
```

```
##       2.5%      97.5%
## 0.04933139 0.40490237
```

```
## [1] 0.9928889
```

```
## [1] 0.1073366
```

```
##        2.5%       97.5%
## -0.09022869  0.29710050
```

```
## [1] 0.9928889
```

```
## [1] -0.2233224
```

```
##        2.5%       97.5%
## -0.44226679 -0.01357876
```

```
## [1] 0.9802222
```

```
## [1] 0.9671667
```

```
## [1] 0.9968333
```

```
## [1] 0.9408889
```

```
## [1] 0.2435904
```

```
##       2.5%      97.5%
## 0.00506527 0.83786956
```

c. Maternal girth

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

```
## [1] 1.75087
```

```
##       2.5%      97.5%
## -0.1581915  3.6946741
```

```
## [1] 0.9627222
```

```
## [1] 0.5389617
```

```
##      2.5%     97.5%
## -1.867172  2.945123
```

```
## [1] 0.672
```

```
## [1] 1.211908
```

```
##     2.5%    97.5%
## -1.20585  3.62550
```

```
## [1] 0.8360556
```

```
## [1] 1.31124
```

```
##       2.5%      97.5%
## -0.5003124  3.1149878
```

```
## [1] 0.9257222
```

```
## [1] -5.056817
```

```
##        2.5%       97.5%
## -10.1877712   0.1005879
```

```
## [1] 0.9735556
```

```
## [1] 0.9995
```

```
## [1] 0.9495
```

```
## [1] 0.7733889
```

```
## [1] 0.6990556
```

```
## [1] 0.2029323
```

```
##       2.5%      97.5%
## 0.06546549 0.43956391
```

d. Cub mass

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

```
## [1] 15.03992
```

```
## [1] 12.61359
```

```
## [1] 10.27669
```

```
## [1] 1.193169
```

```
##     2.5%    97.5%
## 1.078368 1.316664
```

```
## [1] 0.9998889
```

```
## [1] 1.468266
```

```
##     2.5%    97.5%
## 1.270327 1.689124
```

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

```
## [1] 1.231172
```

```
##     2.5%    97.5%
## 1.089171 1.383222
```

```
## [1] 0.9998889
```

```
## [1] 1.302466
```

```
##     2.5%    97.5%
## 1.139436 1.487960
```

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

```
## [1] 1.196805
```

```
##     2.5%    97.5%
## 1.032235 1.379900
```

```
## [1] 0.9915556
```

```
## [1] 1.091756
```

```
##      2.5%     97.5%
## 0.9390356 1.2591582
```

```
## [1] 0.8769444
```

```
## [1] 1.098853
```

```
##     2.5%    97.5%
## 1.027109 1.174189
```

```
## [1] 0.9969444
```

```
## [1] 1.059594
##      2.5%     97.5%
## 0.9693416 1.1558857
## [1] 0.8985556
## [1] 1.038683
##      2.5%     97.5%
## 0.9436946 1.1397700
## [1] 0.7757222
## [1] 0.1908134
##       2.5%      97.5%
## 0.02397041 0.36598753
## [1] 0.9868333
## [1] 0.1384435
##        2.5%       97.5%
## -0.05778246  0.33205612
## [1] 0.9195556
## [1] 0.05236989
##       2.5%      97.5%
## -0.1328365  0.2416246
## [1] 0.7080556
## [1] 0.9378889
## [1] 0.9995556
## [1] 0.9951667
## [1] 0.9997778
## [1] 1
## [1] 0.9973889
## [1] 1.173139
##     2.5%    97.5%
## 1.050121 1.307514
## [1] 1
## [1] 0.9931111
## [1] 0.9818889
## [1] 0.9964444
## [1] 0.9999444
## [1] 0.9978889
## [1] 0.5064444
## [1] 0.08095135
```

```
##       2.5%      97.5%
## 0.02033751 0.18476288
```

e. Litter mass

```
## singleton      twin   triplet
##  15.03992  25.22717  30.83007
```

```
## [1] 1.68056
```

```
##     2.5%    97.5%
## 1.518991 1.854655
```

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

```
## [1] 1.222878
```

```
##     2.5%    97.5%
## 1.084425 1.377194
```

```
## [1] 0.9992222
```

f. Cub survival

```
## [1] 0.9967778
```

```
## [1] 0.3487018
```

```
##       2.5%      97.5%
## 0.03995184 0.63904210
```

```
## [1] 0.9855556
```

```
## [1] 0.351673
```

```
##        2.5%       97.5%
## 0.000958976 0.681853722
```

```
## [1] 0.9755
```

```
## [1] 0.002971179
```

```
##       2.5%      97.5%
## -0.3504390  0.3507407
```

```
## [1] 0.5092222
```

```
## [1] 0.5208333
```

```
## [1] -0.02778369
```

```
## [1] 0.5378333
```

```
## [1] -0.03442667
```

```
## [1] 0.8321667
```

```
## [1] 0.7951667
```

```
## [1] 0.6773333
```

```
## [1] 0.7650556
```

```
## [1] 0.6895556
```

g. Recapture probability

```
## [1] 0.2234877
```

```
##      2.5%     97.5%
## 0.1651372 0.2905332
```

```
## [1] 0.05424124
```

```
##       2.5%      97.5%
## 0.03151335 0.08687671
```

```
## [1] 0.05086043
```

```
##       2.5%      97.5%
## 0.02716686 0.08462128
```

```
## [1] 0.1269051
```

```
##       2.5%      97.5%
## 0.09006323 0.17069503
```

```
## [1] 0.8724887
```

```
##     2.5%    97.5%
## 0.619182 1.224541
```

h. Productivity by litter size

```
## [1] 0.1711728
```

```
##       2.5%      97.5%
## -0.2482271  0.6812235
```

```
## [1] 0.7722222
```

```
## [1] 0.599932
```

```
##      2.5%     97.5%
## 0.1520810 0.9442597
```

```
## [1] 0.9944444
```

```
## [1] 0.2095511
```

```
##       2.5%      97.5%
## -0.2878088  0.7940797
```

```
## [1] 0.7894444
```

```
## [1] 0.2071466
```

```
##       2.5%      97.5%
## -0.8836952  0.9143302
```

```
## [1] 0.8172222
```

```
## [1] -0.2158487
```

```
##       2.5%      97.5%
## -1.1130376  0.9472112
```

```
## [1] 0.6433333
```

```
## [1] -0.8712762
```

```
##       2.5%      97.5%
## -1.7891654 -0.1388365
```

```
## [1] 0.9944444
```

2. Variance explained by model

```
## [1] 0.1249375
```

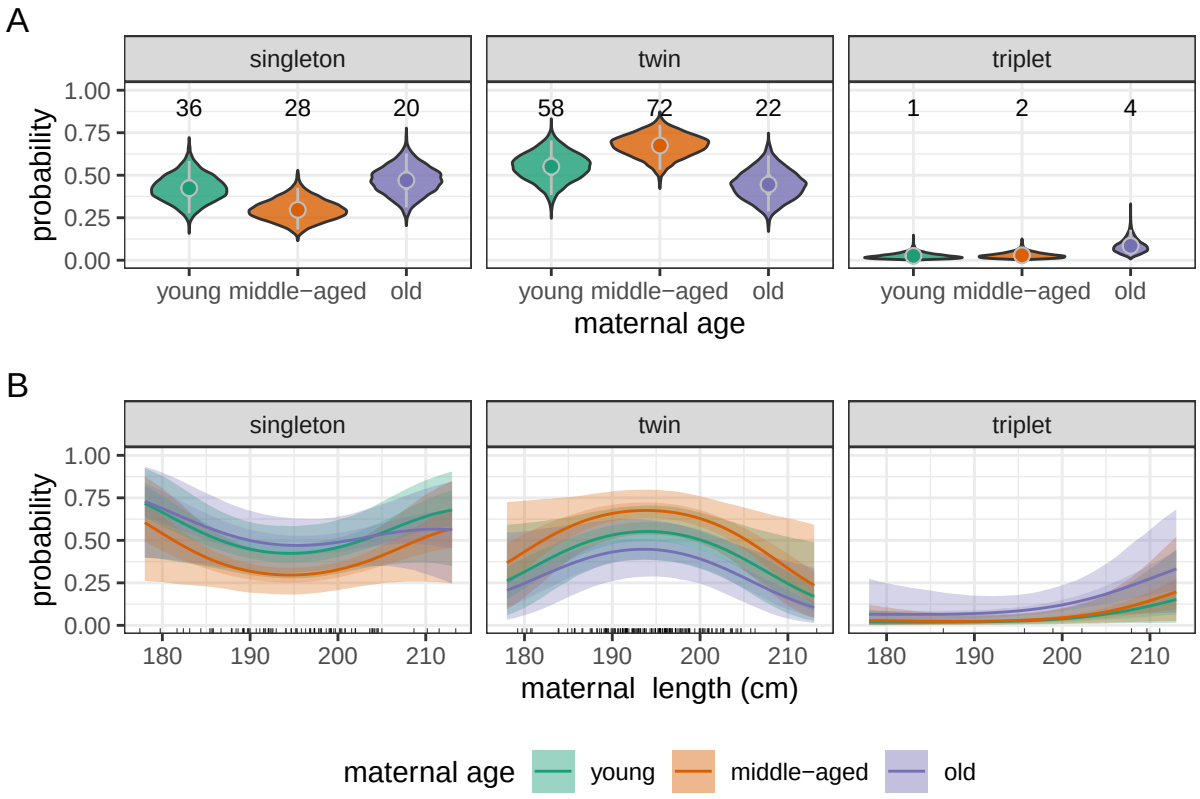
```
## [1] 0.2066141
```

```
## [1] 0.4091651
```

E. 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.
```

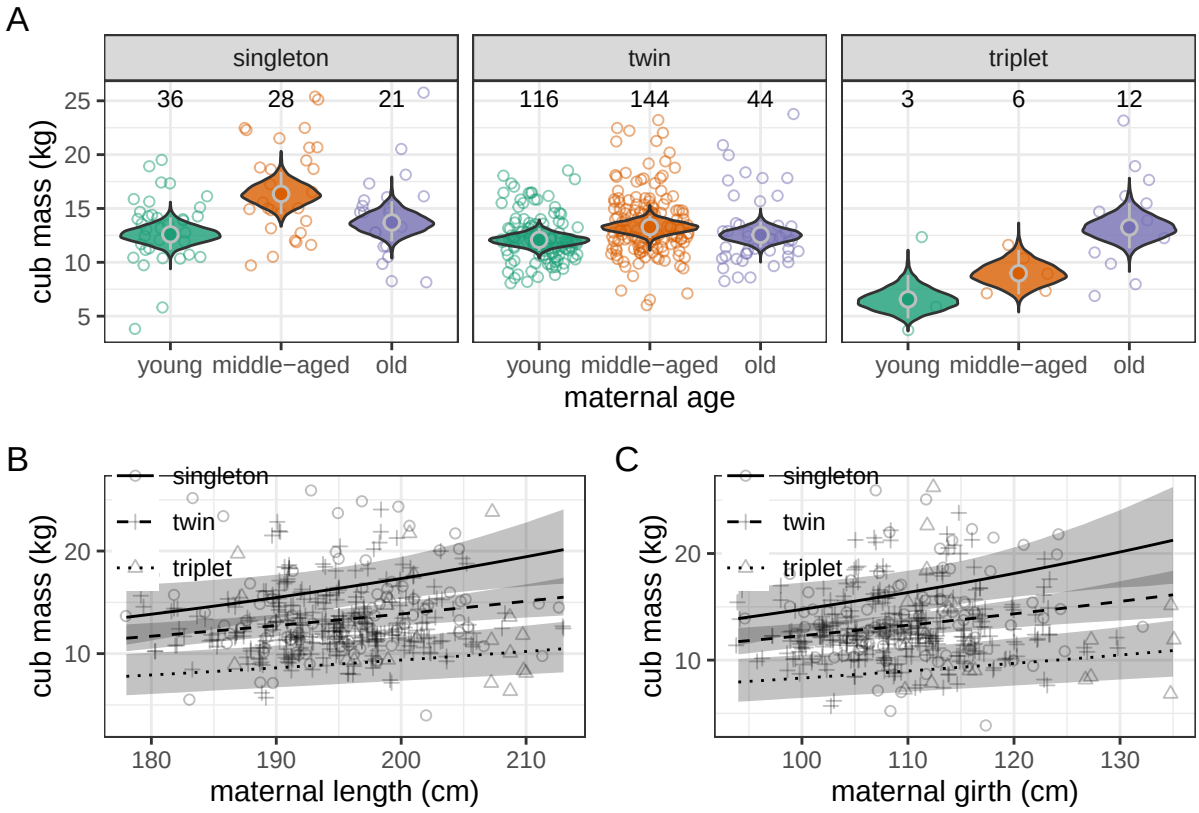


2. Figure 3: Determinants of cub mass

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

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

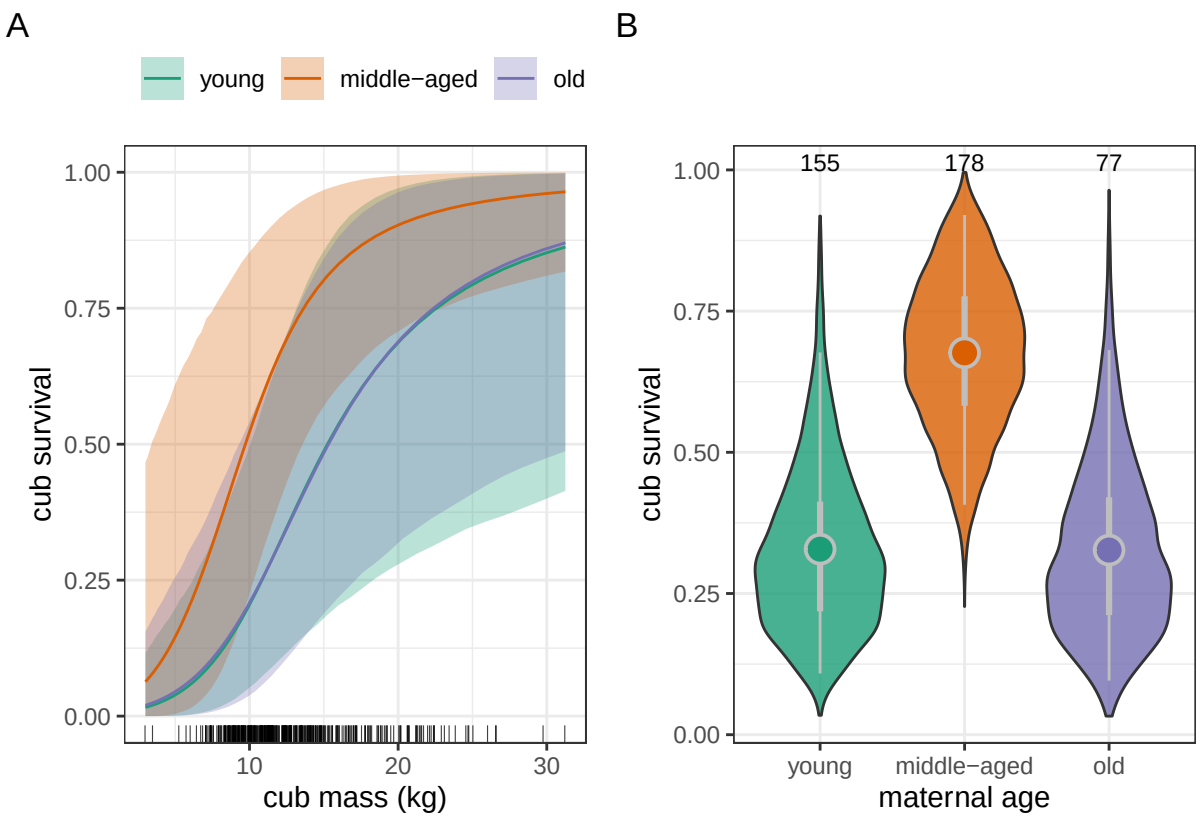
```
## Removed 1 rows containing missing values (‘geom_point()‘).
```




```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).
```

3. Figure 4: Determinants of cub survival

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


A
young
middle−aged
old
1.00
0.75
0.50
0.25
0.00
cub survival
10
20
30
cub mass (kg)
B
155
178
77
1.00
0.75
0.50
0.25
0.00
cub survival
young
middle−aged
old
maternal age

4. Figure 6: Productivity by litter size