The adequacy of aging techniques in vertebrates for rapid estimation of population mortality rates from age distributions

Zhao, M.; Klaassen, C.A.J.; Lisovski, S.; Klaassen, M.

Published in:
Ecology and Evolution

DOI:
10.1002/ece3.4854

License
CC BY

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Appendix S4. Example of R code to simulate data and calculate \( \frac{1}{m \sqrt{I(m)}} \), the basic factor in the 95% error percentage plotted in Fig. 1.

Parameters that will be used in the function

\[ \beta \text{ and } \sigma \text{ is the slope and the standard deviation of the error in equation 19, respectively.} \]

\[ m, \text{ assumed mortality rate.} \]

\[ c, \text{ constant used in calculation of } K(\mu), \text{ discussed in Numerical Computation Fisher Information in Appendix S2.} \]

Construct a function called basic factor to calculate \( \frac{1}{m \sqrt{I(m)}} \), the basic factor in 95% error percentage (EP(95)) in equation 44.

```r
basic_factor <- function(beta = 1, sigma, m, c) {
  beta = abs(beta)
  lambda = -log(1-m) # equation is specified in the line below equation
  mu <- sigma*lambda/beta # equation is specified in the line above equation
  if(mu > 8) {NA} else {
    # integral part of equation 46
    int10 <- function(x) (x + dnorm(x)/pnorm(x)) * dnorm(x + mu)
    int1 <- integrate(int10, -mu - c, -mu + c)$value
    # equation 46 to calculate L(\mu), which is also the lower limit of K(\mu)
    L <- Klow <- mu + int1
    # right hand of equation 47, which is the upper limit of K(\mu)
  }
}
```

# Construct a function called basic factor to calculate \( \frac{1}{m \sqrt{I(m)}} \), the basic factor in 95% error percentage (EP(95)) in equation 44.

basic_factor <- function(beta = 1, sigma, m, c) {
  beta = abs(beta)
  lambda = -log(1-m) # equation is specified in the line below equation
  mu <- sigma*lambda/beta # equation is specified in the line above equation
  if(mu > 8) {NA} else {
    # integral part of equation 46
    int10 <- function(x) (x + dnorm(x)/pnorm(x)) * dnorm(x + mu)
    int1 <- integrate(int10, -mu - c, -mu + c)$value
    # equation 46 to calculate L(\mu), which is also the lower limit of K(\mu)
    L <- Klow <- mu + int1
    # right hand of equation 47, which is the upper limit of K(\mu)
  }
}

# Construct a function called basic factor to calculate \( \frac{1}{m \sqrt{I(m)}} \), the basic factor in 95% error percentage (EP(95)) in equation 44.

basic_factor <- function(beta = 1, sigma, m, c) {
  beta = abs(beta)
  lambda = -log(1-m) # equation is specified in the line below equation
  mu <- sigma*lambda/beta # equation is specified in the line above equation
  if(mu > 8) {NA} else {
    # integral part of equation 46
    int10 <- function(x) (x + dnorm(x)/pnorm(x)) * dnorm(x + mu)
    int1 <- integrate(int10, -mu - c, -mu + c)$value
    # equation 46 to calculate L(\mu), which is also the lower limit of K(\mu)
    L <- Klow <- mu + int1
    # right hand of equation 47, which is the upper limit of K(\mu)
  }
}

# Construct a function called basic factor to calculate \( \frac{1}{m \sqrt{I(m)}} \), the basic factor in 95% error percentage (EP(95)) in equation 44.

basic_factor <- function(beta = 1, sigma, m, c) {
  beta = abs(beta)
  lambda = -log(1-m) # equation is specified in the line below equation
  mu <- sigma*lambda/beta # equation is specified in the line above equation
  if(mu > 8) {NA} else {
    # integral part of equation 46
    int10 <- function(x) (x + dnorm(x)/pnorm(x)) * dnorm(x + mu)
    int1 <- integrate(int10, -mu - c, -mu + c)$value
    # equation 46 to calculate L(\mu), which is also the lower limit of K(\mu)
    L <- Klow <- mu + int1
    # right hand of equation 47, which is the upper limit of K(\mu)
  }
}
Kup <- L + (1 + 1/c + 1/c^2) * dnorm(c)

# use the average of the upper and lower limit to represent $K(\mu)$ in equation 47
K <- (Kup + Klow)/2

Jmu <- 1/(mu^2) - 1 - mu^2 + mu*K # second equation in equation 45

Im <- (sigma/beta/(1-m))^2*Jmu # equation 38

mIm <- 1/(sqrt(Im)*m) # the basic factor in equation 44 and plotted in Fig. 1

# return $\frac{1}{m\sqrt{I(m)}}$ in equation 44
if(is.infinite(mIm)|is.nan(mIm)) {
    NA # when mu > 8 the computation is unreliable and thus discarded
} else {
    mIm
}

}