



W&M ScholarWorks

Reports

10-2016

Simulations of the effect of sample size on the maximum observed age in a fish population

John M. Hoenig
Virginia Institute of Marine Science

Follow this and additional works at: <https://scholarworks.wm.edu/reports>



Part of the [Aquaculture and Fisheries Commons](#), and the [Marine Biology Commons](#)

Recommended Citation

Hoenig, J. M. (2016) Simulations of the effect of sample size on the maximum observed age in a fish population. Data report (Virginia Institute of Marine Science); no. 64. Virginia Institute of Marine Science, William & Mary. <https://doi.org/10.21220/V5HB40>

This Report is brought to you for free and open access by W&M ScholarWorks. It has been accepted for inclusion in Reports by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.

Simulations of the effect of sample size on the maximum observed age in a fish population

John M. Hoenig

Department of Fisheries Science
Virginia Institute of Marine Science
The College of William and Mary

Data Report 64

November 2016

doi:10.21220/V5HB40



This technical report provides the R computer code for performing the Monte Carlo simulations described in the publication:

Hoenig, J.M. (2017). Should Natural Mortality Estimators Based on Maximum Age Also Consider Sample Size? Transactions of the American Fisheries Society. 146: 136-146.

doi: 10.1080/00028487.2016.1249291

simulating the maximum age in samples of size n drawn from

a population in equilibrium with constant recruitment and

constant mortality rate with respect to age and time

This simulation is in support of the article by John Hoenig

which examines several expressions in the literature for the

expected value of the maximum age in a sample drawn at one

point in time. These expressions were derived by Sarhan (1954),

Holt (1965), Hoenig (1983), and Kenchington (2014).

Janos (John) Hoenig rev Sept 11, 2016

Virginia Institute of Marine Science, Gloucester Pt., Virginia

hoenig@vims.edu

```

# generate growth of a population from time 0 to 200 assuming
# constant recruitment and mortality starting at time 0. (For
# Z = .2, the population approaches steady state after about 60
# years. Hence, we can sample the population at time 200, say,
# and have approximately equilibrium conditions (although,
# strictly speaking, the oldest age can't be more than 200))

# note: I reset the random number seed several times to
# prevent recycling of the number stream

# Generate 40 million times of birth distributed uniformly in
# the interval (0,200)
set.seed(45)
birth.times <- runif(20000000, 0, 200)
set.seed(23)
birth.times <- c(birth.times, runif(20000000, 0, 200))

# generate a corresponding duration of life for each birth
# from an exponential distribution with mortality rate Z
set.seed(14)
Z = .2 # instantaneous mortality rate (hazard function), i.e., 1/mean
life.dur <- rexp(20000000, Z)
set.seed(17)
life.dur <- c(life.dur, rexp(20000000, Z))
max(life.dur) # longest life among the 40 million lives generated: [1] 98.91561

#####

# quality control checks

```

```
1/mean(life.dur) # Check mean is consistent with Z, i.e., close to 1/Z: [1] 0.2000247
```

```
# check on distribution of life durations
```

```
opar <- par(mfrow=c(2,2))
```

```
out <- hist(life.dur, main="life durations", xlab="duration of life")
```

```
plot(out$breaks[1:(length(out$breaks)-1)], log(out$counts), ylab="log(Frequency)",  
      xlab="duration of life")
```

```
out2 <- hist(life.dur[life.dur>70], main="life durations > 70", xlab="duration of life")
```

```
plot(out2$breaks[1:(length(out2$breaks)-1)], log(out2$counts), ylab="log(Frequency)",  
      xlab="duration of life")
```

```
par(opar)
```

```
# slight departure from linearity on the log scale at the extreme right
```

```
# is due to fact that one can't observe any animal abundances between
```

```
# 0 and 1, hence the linear regression can't extend below the x-axis
```

```
# which begins at  $y = \log(1 \text{ animal}) = 0$ 
```

```
#####
```

```
annual.recruit <- 40000000/200 ; annual.recruit # 200,000
```

```
# compute time of death for each fish as birth time + length of life
```

```
death.times <- birth.times + life.dur
```

```
# look at the population that is alive at time 200
```

```
time <- 200
```

```
# find all animals alive at time=time
```

```
alive.t <- birth.times[birth.times<time & time < death.times]
```

```
length(alive.t) # size of the population # [1] 999470
```

```
#####
```

```
# quality control checks
```

```
# look at the population that is alive as a function of time to
```

```
# see the evolution of the population ### THIS RUNS VERY SLOWLY
```

```
pop <- rep(NA, 200)
```

```
for (time in 1:200){
```

```
# find all animals alive at time=time
```

```
  alive.t <- birth.times[birth.times<time & time < death.times]
```

```
  pop[time] <- length(alive.t) # size of the population
```

```
}
```

```
plot(1:200, pop, typ="l", xlab="time", ylab="population size")
```

```
plot(50:200, pop[50:200], typ="l", xlab="time", ylab="population size")
```

```
# population grows steadily towards an equilibrium value and has largely
```

```
# reached the equilibrium after 50 years. There is not much variability
```

```
# in abundance after 50 years.
```

```
#####
```

```
# make histogram of the times of birth for those animals alive
```

```
# at time=200 ##### THIS CREATES FIGURE 1 for the paper #####
```

```
opar <- par(las=1, cex.axis=1.2)
```

```
out <- hist(alive.t, axes=F, main="", col=1,
```

```
  xlab=list("Birth time", cex=1.5), ylab = list("Frequency (thousands)",cex=1.5))
```

```
axis(2, at= seq(0, 600000, 100000), labels = seq(0, 600, 100), las=1)
```

```
axis(1, at = out$breaks, labels = out$breaks)
```

```
box()
```

```
ages <- time - alive.t
```

```
max(ages) # find the oldest animal alive at time=time: [1] 62.97065
```

```

hist(ages, las=0)

box()

par(opar)

#####

# Now let's look at how maximum age varies with changes in sample size.
# We will draw samples of size n (= 100, 500 and 1000) 100,000 times
### NOTE: THIS IS VERY SLOW ###

obs.max.age <- matrix(NA, nrow=100000, ncol = 3) # hold the 300,000 maximum ages

### THIS WILL BE SLOW ###

n <- c(100, 500, 1000)
for (i in 1:3) {
  for (j in 1:100000){
    samp <- sample(ages, n[i])    # draw a sample
    obs.max.age[j, i] <- max(samp) # note the maximum age in sample
  }
}

mean.obs.max <- apply(obs.max.age, 2, mean) # [1] 25.87964 34.01604 37.62368

# NOTE: the use of the nested loops above is obviously inefficient but
# I perform the computations this way to make them as easy for the reader
# to follow as possible

# now compute the expected values for each sample size using the equations
# given by Johnson and Kotz (called expec), Hoenig (called Hoe), Holt (called Holt)
# and Kenchington (called Ken). Also compute the variance by the formula of
# Johnson and Kotz (called varexpec)

expec <- rep(NA, 3)

```

```

Hoe <- rep(NA, 3)
Holt <- rep(NA, 3)
Ken <- rep(NA, 3)
varexpec <- rep(NA, 3)

for (i in 1:3) {
  ind <- 1:n[i]
  expec[i] <- 1/(Z) * sum(1/ind)
  Hoe[i] <- log(2 * n[i] + 1)/Z
  Holt[i] <- (.577 + log(n[i]))/Z
  Ken[i] <- log(2 * n[i] * Z + 1)/Z
  varexpec[i] <- (1/Z) * sum(1/ind^2)
}

sqrt(varexpec)

answer = cbind(n, round(mean.obs.max, 2), round(expec, 2), round(Ken, 2), round(Holt, 2), round(Hoe,
2))
colnames(answer) <- c("n", "obs", "expect", "Kench", "Holt", "Hoe appr")
answer
#   n  obs expect Kench Holt Hoe appr
# [1,] 100 25.90 25.94 18.57 25.91 26.52
# [2,] 500 34.05 33.96 26.52 33.96 34.54
# [3,] 1000 37.59 37.43 29.97 37.42 38.01

##### THIS CREATES FIGURE 2 #####

opar = par(las=1, cex.axis=1.2)
plot(n, mean.obs.max, typ="l", ylim=c(17, 38), xlab=list("Sample size", cex=1.5),

```



```

      ylab=list("Mean oldest age", cex=1.5), lty=3, lwd=2)
points(n, expec, typ="l", lty=1, lwd=2)
points(n, Ken, typ="l", lty=4, lwd=2)
# points(n, Holt, typ="l", lty=5, pch=5, lwd=2)
points(n, Hoe, typ="l", lty=2, lwd=2)

leg <- c("Hoenig approx.", "observed", "Sarhan", "Kenchigton" )
text(800, 23.5, "top to bottom")
legend("bottomright", legend=leg, lty = c(2,3,1,4), bty="n")
#title("maximum age when M = 0.2")

par(opar)

#####

# Now compare Kenchington's formula (called ken) to the one in
# Johnson and Kotz (called true) for values of mortality rate
# varying from 0.05 to 1.5, for 3 sample sizes

##### THIS CREATES FIGURE 3 #####
Zvec <- seq(.05, 1.5, .1) ; Zvec
n <- c(100, 500, 1000)

compare <- function(Z, n) { # compute E(Tmax) from Sarhan &
      # Kenchington formulae
      true <- sum(1/1:n)/Z
      ken <- log(2 * n * Z + 1)/Z
      return(c(true, ken))
}

```

```

results100 <- t(sapply(Zvec, compare, 100))
results500 <- t(sapply(Zvec, compare, 500))
results1000 <- t(sapply(Zvec, compare, 1000))

# first do n = 1000
wid <- 2
opar = par(las=1, cex.axis=1.2, fig=c(0,1,0,1), bty="o")
plot(Zvec, results1000[,1], typ="l", lty=1, lwd=wid, xlab=list("Mortality rate", cex=1.5),
      ylab=list("Maximum age", cex=1.5))
points(Zvec, results1000[,2], typ="l", lty=2, lwd=wid)

leg <- c("correct, n=1000", "Kenchington, n=1000")
legend("topright", leg, lty=c(1,2), bty="n")

par(new=T, fig=c(x1=.271, x2=.991, y1=.25, y2=.85), bty="l")
plot(Zvec[6:15], results1000[6:15,1], typ="l", lty=1, lwd=wid, xlab="",
      ylab="", ylim=c(4,14), yaxt="n")
points(Zvec[6:15], results1000[6:15,2], typ="l", lty=2, lwd=wid)
axis(2, at = seq(4, 14, 2), labels = c("4", "", "8", "", "12", ""))

par(opar)

# now do n= 100
wid <- 2
opar = par(las=1, cex.axis=1.2, fig=c(0,1,0,1))
plot(Zvec, results100[,1], typ="l", lty=1, lwd=wid, xlab=list("Mortality rate", cex=1.5),
      ylab=list("Maximum age", cex=1.5), bty="o")
points(Zvec, results100[,2], typ="l", lty=2, lwd=wid)

```

```

leg <- c("correct, n=100", "Kenchington, n=100")
legend("topright", leg, lty=c(1,2,3,4), bty="n")

par(new=T, fig=c(x1=.271, x2=.991, y1=.25, y2=.85), bty="l")
plot(Zvec[6:15], results100[6:15,1], typ="l", lty=1, lwd=wid, xlab="",
     ylab="", ylim=c(4,14), yaxt="n")
points(Zvec[6:15], results100[6:15,2], typ="l", lty=2, lwd=wid)
axis(2, at = seq(4, 14, 2), labels = c("4","","8","","12",""))
par(opar)

#####

# Annual (not continuous) reproduction case

set.seed(45)
Z = .2

# We generate birth times up to time 201 so that when we truncate the
# birth time to an integer we will have some animals born at time 200,
# in other words, the age composition at time = 200 will include animals
# of age 0
birth.times <- floor(runif(20000000, 0, 201))
set.seed(23)
birth.times <- c(birth.times, floor(runif(20000000, 0, 201)))
range(birth.times) # this gives births ranging from time 0 to 200
hist(birth.times)

# the first bar of the histogram is higher than the others because
# it represents the animals born from year 0 to year 10 (11 years)

```

```

# while all the other bars represent 10 years of births (11 to 20,
# 21 to 30, 31 to 40, etc.)

set.seed(14)
life.dur <- rexp(20000000, Z)
set.seed(17)
life.dur <- c(life.dur, rexp(20000000, Z))
max(life.dur)

1/mean(life.dur) # Check that mean is consistent with a Z of 0.5
death.times <- birth.times + life.dur

# Look at the population at time 200
time <- 200
alive.t <- birth.times[birth.times<=time & time < death.times]
length(alive.t) # size of the population
hist(alive.t, main="birth times for fish alive at time=200",
      xlab=list("Birth time", cex=1.5), ylab = list("Frequency",cex=1.5))
ages <- time - alive.t
max(ages) # [1] 63

# Now let's look at how maximum age varies with changes in sample size
#### THIS RUNS SLOWLY
obs.max.age <- matrix(NA, nrow=100000, ncol = 3)

n <- c(100, 500, 1000)
for (i in 1:3) {
  for (j in 1:100000){
    samp <- sample(ages, n[i])

```

```

    obs.max.age[j, i] <- max(samp)
  }
}

mean.obs.max <- apply(obs.max.age, 2, mean)

expec <- rep(NA, 3)
Hoe <- rep(NA, 3)
Ken <- rep(NA, 3)
Holt <- rep(NA, 3)
for (i in 1:3) {
  ind <- 1:n[i]
  expec[i] <- 1/(Z) * sum(1/ind)
  Hoe[i] <- log(2 * n[i] + 1)/Z
  Ken[i] <- log(2 * n[i] * (1 - exp(-Z)) + 1)/Z
  Holt[i] <- (0.577 + log(n[i]))/Z
}

answer = cbind(n, round(mean.obs.max, 2), round(expec, 2), round(Ken, 2), round(Hoe, 2),
              round(Holt,2))
colnames(answer) <- c("n", "obs", "expect", "Kench", "Hoe appr", "Holt") ; answer

##### THIS CREATES FIGURE 4 #####
par(las=1, cex.axis=1.2, bty="o")
plot(n, Hoe, typ="l", ylim=c(18, 40), xlab=list("Sample size", cex=1.5),
     ylab=list("Mean oldest age", cex=1.5), lty=2, lwd=2)
points(n, expec, typ="l", lty=1, lwd=2)
points(n, mean.obs.max, typ="l", lty=3, lwd=2)
points(n, Ken, typ="l", lty=4, lwd=2)

```

```
# points(n, Holt, typ="b", lty=5, pch=5, lwd=2, col=2) # Holt is on top of expec
leg <- c("Hoenig approx.", "expected (expon)", "observed", "Kenchington")
legend("bottomright", legend=leg, lty = c(2, 1, 3, 4), bty="n")
text(790, 24.7, "top to bottom")
par(opar)
```

```
#####
```