Package ‘SAFR’

November 28, 2016

Type Package

Title Survival Analysis for Fisheries Research demonstrates applications of survival analysis (Cox, 1984) to analyze catch at age data in fisheries research

Version 0.1

Date 2014-10-27

Author Marco Kienzle

Maintainer Marco Kienzle <Marco.Kienzle@gmail.com>

Description This package implement several likelihood function that allow the user to estimate mortality rates (both natural and fishing) as well as selectivity and recruitment from catch at age data from fisheries.

R topics documented:

Caaa2Coaa 2
Coaa2Caaa 2
EstimateMandQ 3
EstimateRecruitment 4
EstimateZ 5
GenerateData2 6
llfunc1 6
llfunc2 7
llfunc3 9
llfunc4 10
llfunc5 12
llfunc7 13
logistic 14
my.cumsum 14
total.over.lines 15
which.cohort 15

Index **16**

Caaa2Coaa *Convert catch-at-age to cohort-at-age*

Description

Convert a matrix of dimensions year x age into a matrix of cohort x age

Usage

```
Caaa2Coaa(mat)
```

Arguments

mat	a matrix of catch at age

References

NONE

Examples

```
nb.at.age <- matrix(sample(1:10, 40, replace = TRUE), nrow = 10, ncol = 4)
Caaa2Coaa(nb.at.age)
```

Coaa2Caaa *Convert cohort-at-age to catch-at-age - the opposite of Caaa2Coaa*

Description

Convert a matrix of dimensions number of cohorts x age into a matrix of year x age

Usage

```
Coaa2Caaa(cohort.mat)
```

Arguments

cohort.mat	a matrix of cohort at age

References

NONE

Examples

```
(nb.at.age <- matrix(sample(1:10, 40, replace = TRUE), nrow = 10, ncol = 4))
tmp <- Caaa2Coaa(nb.at.age)
Coaa2Caaa(tmp)
```

EstimateMandQ *Estimate total mortality (Z) using catch at age from a single cohort*

Description

This function provide an estimate of total mortality (Z) by maximum likelihood using a vector of catch at age from a cohort for each age group from 0 to max.age

Usage

```
EstimateMandQ(catch, effort, catchability.scaling.factor)
```

Arguments

`catch`	is a vector of numerics describing the number of fish caught in each yearly age-group. The first value reprensent fishes caught with age between 0 and 1 year old
`effort`	is a vector of numerics describing fishing effort in any specific year
`catchability.scaling.factor`	A factor to scale the parameters during the optimization

References

Quinn and Deriso (1999) - Quantitative Fish Dynamics

Examples

```
# Suppose age varies between 0 and 10
age <- seq(0,10)

# Generate a random natural mortality
M <- runif(1, min = 1e-2, max = 0.3)

effort <- runif(length(age)-1, min = 1e3, max = 2e3)
catchability <- runif(1, min = 1/3e3, max = 1/2e3)

# Catchability scaling factor
csf <- 1e-4
F <-  catchability * effort

print(paste("Simulated q is", round(catchability / csf,3), as.character(csf)))
print(paste("Simulated M is ", round(M,3)))

N0 <- runif(1, min = 4e3, max = 1e4)
print(paste("Simulated recruitment is", round(N0)))

# Calculate number at age using a simple exponential model ( see Quinn and Deriso, 1999)
nb.at.age <- cbind(age, N0 * exp(-c(0, cumsum(M + F))))

# Calculate the total number of individual dying at age
total.death <- N0 * (exp(-c(0,cumsum(M+F)[-length(effort)])) - exp(-cumsum(M+F)))

# Number of fish dying from fishing is a fraction of total mortality
```

```
catch <- F/(M+F) * total.death

# Estimate q and M
best.qM.est <- EstimateMandQ(catch, effort, catchability.scaling.factor
= csf)

errors <- sqrt(diag(solve(best.qM.est$hessian)))

print(" ##### ")
print(paste("Estimated catchability is", round(best.qM.est$par[1],3), "+-", round(errors[1],3), as.character
print(paste("Estimated M is", round(best.qM.est$par[2],3), "+-", round(errors[2],3)))
```

EstimateRecruitment — *Estimate recruitment using total mortality (Z) and catch at age from a single cohort*

Description

This function provide an estimate of recruitment using total mortality (Z) and catch at age. The idea is that each catch at age figure provides an estimate of recruitment which first and second moment are calculated

Usage

```
EstimateRecruitment(Z)
```

Arguments

Z	a numerica scalar >= 0

References

no reference

Examples

```
# Suppose age varies between 0 and 10
age = seq(0,10)

# Suppose you have a M=0.105, F and N0 are arbitrary (randomly generated)
M <- 0.105
F <- runif(1, min = 0.1, max = 3)
print(paste("Simulated Z is", round(M+F,3)))

# Generate a random recruitment
N0 <- runif(1, min = 1e3, max = 1e4)
print(paste("Simulated recruitment", round(N0)))

# Calculate number at age using a simple exponential model ( see Quinn and Deriso, 1999)
nb.at.age <- cbind(age, N0 * exp(-(M+ F)) ^ age)

# Calculate the total number of individual dying at age
total.death <- N0 * (exp(-(M+F) * age) - exp(-(M+F) * (age+1)))
```

```
# And the fraction dying from fishing
catch <- F/(M+F) * total.death

# Estimate Z
#best.Z.estimate <- EstimateZ(catch)
#best.Rec.estimate <- EstimateRecruitment(Z=best.Z.estimate$par)
```

EstimateZ *Estimate total mortality (Z) using catch at age from a single cohort*

Description

This function provide an estimate of total mortality (Z) by maximum likelihood using a vector of catch at age from a cohort for each age group from 0 to max.age

Usage

```
EstimateZ(catch)
```

Arguments

`catch`	is a vector of numerics describing the number of fish caught in each yearly age-group. The first value reprensent fishes caught

References

Quinn and Deriso (1999) - Quantitative Fish Dynamics

Examples

```
# Suppose age varies between 0 and 10
age = seq(0,10)

# Suppose you have a M=0.105, F and N0 are arbitrary (randomly generated)
M <- 0.105
F <- runif(1, min = 0.1, max = 3)
print(paste("Simulated Z is", round(M+F,3)))

# Generate a random recruitment
N0 <- runif(1, min = 1e3, max = 1e4)
print(paste("Simulated recruitment", round(N0)))

# Calculate number at age using a simple exponential model ( see Quinn and Deriso, 1999)
nb.at.age <- cbind(age, N0 * exp(-(M+ F)) ^ age)

# Calculate the total number of individual dying at age
total.death <- N0 * (exp(-(M+F) * age) - exp(-(M+F) * (age+1)))

# And the fraction dying from fishing
catch <- F/(M+F) * total.death

# Estimate Z
best.Z.est <- EstimateZ(catch)
```

GenerateData2 — *Convert catch-at-age to cohort-at-age*

Description

Convert a matrix of year x age into a matrix of cohort x age

Usage

```
GenerateData2(max.age = 10, nb.of.cohort = 20, ...)
```

Arguments

`max.age`	max age
`nb.of.cohort`	number of cohorts
`...`	other parameters

References

NONE

llfunc1 — *log-likelihood function of catch at age and total mortality (Z) written to Z*

Description

This function calculate the log-likelihood of the survival model assuming constant mortality rate (Z) given catch at age.

NOTE that the optimization does not constrains Z > 0 because we noticed problems of convergence of the example below on 32-bit systems when using method = "L-BFGS-B".

Usage

```
llfunc1(Z, catch, plus.group)
```

Arguments

`Z`	a positive or null scalar giving the constant mortality rate (units 1/year)
`catch`	a vector of number of individual caught at age. The first value represent a number of animal between age 0 and 1, the second between 1 and 2, etc...
`plus.group`	a boolean indicating whether the data for the last age-group is the sum of observation for this age-group and all olders one or not

References

Cox (1984) - Analysis of survival data Dupont (1983) - A Stochastic catch-effort method for estimating animal abundance, Biometrics 39, 1021–1033 Chiang (1968) - Introduction to stochastic processes in biostatistics, John Wiley & Sons

Examples

```
# Suppose age varies between 0 and 10
age <- seq(0,10)

# Suppose you have a M=0.105, F and N0 are arbitrary (randomly generated)
M <- 0.105
F <- runif(1, min = 0.1, max = 3)
print(paste("Simulated Z is", round(M+F,3)))

# Generate a random recruitment
N0 <- runif(1, min = 1e3, max = 1e4)
print(paste("Simulated recruitment", round(N0)))

# Calculate number at age using a simple exponential model ( see Quinn and Deriso, 1999)
nb.at.age <- cbind(age, N0 * exp(-(M+ F)) ^ age)

# Calculate the total number of individual dying at age
total.death <- N0 * (exp(-(M+F) * age) - exp(-(M+F) * (age+1)))

# And the fraction dying from fishing
catch <- F/(M+F) * total.death

# Estimate Z
result <- optim(par = c(0.1), fn = llfunc1, catch = catch, method = c("L-BFGS-B"),
      lower = c(1e-2), upper = c(3), hessian = TRUE)
print(paste("Estimated Z is", round(result$par,3), "+-", round(sqrt(diag(solve(result$hessian))),3)))

### Estimate of Z using a plus group
ap <- ifelse( (0.0001 * sum(catch)) < catch[11], 11, min(which(catch < (0.0001 * sum(catch)) )))
catch1 <- catch[1:ap]; catch1[ap] <- sum(catch[ap:11]) # create the +group

# if the number of observation is large enough so that there is no need to create a +group
# then do not use the +group option
ifelse( length(catch) == length(catch1), print("+group option not used"),
{
result1 <- optim(par = c(0.1), fn = llfunc1, catch = catch1, plus.group = TRUE, method = c("L-BFGS-B"),
      lower = c(1e-2), upper = c(3), hessian = TRUE)
print(paste("Estimated Z is", round(result1$par,3), "+-", round(sqrt(diag(solve(result1$hessian))),3)))
})

print("# An estimate of recruitment")
# According to Dupont (1983) and Chiang (1968) cohort abundance at t=0 can be estimated by the ratio of total n

#rec.est <- catch / ((result$par[1] - M)/result$par[1]) / (exp(-result$par[1] * #age) - exp(-result$par[1] *
#print(mean(rec.est))
rec.est <- sum(catch) / ((result$par[1] - M)/result$par[1]) / sum( (exp(-result$par[1] * age) - exp(-result$p
print(rec.est)
print(paste("Compared to N0=", round(N0,3)))
```

`llfunc2` *log-likelihood function of catch at age to estimate catchability and natural mortality*

Description

This function calculate the log-likelihood of the survival model assuming fishing mortality is a linear function of effort and natural mortality is constant.

NOTE that the optimization does not constrains both q and M to be > 0 because we noticed problems of convergence of the example below on 32-bit systems when using method = "L-BFGS-B".

Usage

```
llfunc2(par, catch, effort, catchability.scaling.factor)
```

Arguments

`par`	a vector of two parameters: catchability and natural mortality
`catch`	a vector of catch
`effort`	a vector of effort
`catchability.scaling.factor`	a factor to scale the parameters

References

Cox (1984) - Analysis of survival data

Examples

```
# Suppose age varies between 0 and 10
age <- seq(0,10)

# Generate a random natural mortality
M <- runif(1, min = 1e-2, max = 0.3)

effort <- runif(length(age)-1, min = 1e3, max = 2e3)
catchability <- runif(1, min = 1/3e3, max = 1/2e3)

# Catchability scaling factor
csf <- 1e-4
F <-  catchability * effort

print(paste("Simulated q is", round(catchability / csf,3), as.character(csf)))
print(paste("Simulated M is ", round(M,3)))

N0 <- runif(1, min = 4e3, max = 1e4)
print(paste("Simulated recruitment is", round(N0)))

# Calculate number at age using a simple exponential model ( see Quinn and Deriso, 1999)
nb.at.age <- cbind(age, N0 * exp(-c(0, cumsum(M + F))))

# Calculate the total number of individual dying at age
total.death <- N0 * (exp(-c(0,cumsum(M+F)[-length(effort)])) - exp(-cumsum(M+F)))

# Number of fish dying from fishing is a fraction of total mortality
catch <- F/(M+F) * total.death

# Estimate q and M
best.qM.est <- optim(par = c(10,1), fn = llfunc2, catch = catch, effort
```

```
= effort, catchability.scaling.factor = csf, hessian = TRUE)

errors <- sqrt(diag(solve(best.qM.est$hessian)))

### Estimate recruitment
# According to Dupont (1983) Biometrics vol. 39 No 4 pp. 1021-1033

est.rec <- sum(catch)/sum(prob.for.llfunc2(best.qM.est$par, effort, csf))

## And not finding a better way to calculate the uncertainty
ind.rec <- catch / prob.for.llfunc2(best.qM.est$par, effort, csf)
#est.rec.limits <- c("Lower" = sum(catch)/sum(prob.for.llfunc2(best.qM.est$par - errors, effort, csf)), "Upp

print(" ##### ")
print(paste("Estimated catchability is", round(best.qM.est$par[1],3), "+-", round(errors[1],3), as.character
print(paste("Estimated M is", round(best.qM.est$par[2],3), "+-", round(errors[2],3)))
print(paste("Estimated recruitment is", round(est.rec,0), " ranging from ", round(min(ind.rec),0), " to ", ro
```

`llfunc3` *log-likelihood function of catch at age matrix to estimate catchability, selectivity and natural mortality from a matrix of catch at age*

Description

This function calculate the log-likelihood of the survival model assuming fishing mortality the outer product of catchability times effort and selectivity and natural mortality is constant.

Usage

```
llfunc3(par, catch, effort, selectivity.at.age, catchability.scaling.factor,plus.group)
```

Arguments

`par`	a vector of two parameters: catchability and natural mortality
`catch`	a matrix containing number at age in the catch
`effort`	a matrix of effort
`selectivity.at.age`	a vector of selectivity at age bound between 0 and 1
`catchability.scaling.factor`	a factor to scale the parameters
`plus.group`	a boolean indicating whether the data for the last age-group is the sum of observation for this age-group and all olders one or not

References

NONE

Examples

```
# First example, estimate mortality rates assuming selectivity known exactly
set.seed(3)
max.age <- 9
nb.of.cohort <- 30
population <- GenerateData2(max.age = max.age, nb.of.cohort = nb.of.cohort) # Generate catch using gear selec

# sample a fix number of fish each years
n.sample.per.year <- 1e3
nb.at.age.sample <- draw.sample(population$catch, sample.size = n.sample.per.year * (nb.of.cohort + 1 - max.a


# Estimate assuming you know selectivity
result <- optim(par = c(0.2,1), fn = llfunc3, catch = nb.at.age.sample, effort = population$effort, catchabil

errors <- sqrt(diag(solve(result$hessian)))

print(paste("Estimated catchability is", round(result$par[1],3), "+-", round(errors[1],3), " x 10^-4"))
print(paste("Estimated M is", round(result$par[2],3), "+-", round(errors[2],3)))


# Second example to show how to use a +group
set.seed(12)
max.age <- 25
nb.of.cohort <- 75
population <- GenerateData2(max.age = max.age, nb.of.cohort = nb.of.cohort) # Generate catch using gear selec

# sample a fix number of fish each years
n.sample.per.year <- 1e3
nb.at.age.sample <- draw.sample(population$catch, sample.size = n.sample.per.year * (nb.of.cohort + 1 - max.a

# Estimate assuming you know selectivity
result <- optim(par = c(0.2,1), fn = llfunc3, catch = nb.at.age.sample, effort = population$effort, catchabil

errors <- sqrt(diag(solve(result$hessian)))

print(paste("Estimated catchability is", round(result$par[1],3), "+-", round(errors[1],3), " x 10^-4"))
print(paste("Estimated M is", round(result$par[2],3), "+-", round(errors[2],3)))

# Using a +group
ap <- 20
nb.at.age.sample2 <- nb.at.age.sample[,1:ap]; nb.at.age.sample2[,ap] <- rowSums(nb.at.age.sample[,ap:max.age
effort2 <- population$effort[,1:ap];

result2 <- optim(par = c(0.2,1), fn = llfunc3, catch = nb.at.age.sample2, effort = effort2, catchability.scal

errors2 <- sqrt(diag(solve(result2$hessian)))

print(paste("Estimated catchability is", round(result2$par[1],3), "+-", round(errors2[1],3), " x 10^-4"))
print(paste("Estimated M is", round(result2$par[2],3), "+-", round(errors2[2],3)))
```

llfunc4 *log-likelihood function of catch at age matrix to estimate catchability, selectivity and natural mortality*

Description

This function calculate the log-likelihood of the survival model assuming fishing mortality the outer product of catchability times effort and selectivity and natural mortality is constant.

Usage

```
llfunc4(par, catch, effort, catchability.scaling.factor,plus.group)
```

Arguments

`par`	a vector of parameters: catchability, natural mortality and 1 gear selectivity for each age-group
`catch`	a matrix containing number at age in the catch
`effort`	a matrix of effort
`catchability.scaling.factor`	a factor to scale the parameters
`plus.group`	a boolean indicating whether the data for the last age-group is the sum of observation for this age-group and all olders one or not

References

NONE

Examples

```
# without a function for gear selectivity, it is very difficult (impossible) to estimate parameters of interes
# we need substantially more data
# Simulate data
set.seed(3)
max.age <- 9
nb.of.cohort <- 50
population <- GenerateData2(max.age = max.age, nb.of.cohort = nb.of.cohort) # Generate catch using gear selec

# sample a fix number of fish each years
n.sample.per.year <- 2e3
nb.at.age.sample <- draw.sample(population$catch, sample.size = n.sample.per.year * (nb.of.cohort + 1 - max.a

# Estimate parameters
result2 <- optim(par = c(0.2,1, c(rep(1e-12,5), rep(1, max.age-5))), fn = llfunc4, catch = nb.at.age.sample, 
     lower = c(5e-2,5e-2, rep(1e-12, max.age)), upper = c(10,0.5, rep(1, max.age)), hessian = TRUE, control =
errors2 <- sqrt(abs(diag(solve(result2$hessian))))

print(cbind("Estimate" = result2$par, "Error" = errors2))
```

llfunc5 *log-likelihood function of catch at age matrix to estimate catchability, selectivity and natural mortality*

Description

This function calculate the log-likelihood of the survival model assuming fishing mortality the outer product of catchability times effort and selectivity and natural mortality is constant.

Usage

```
llfunc5(par, catch, effort, catchability.scaling.factor)
```

Arguments

`par` a vector of two parameters: catchability and natural mortality

`catch` a matrix containing number at age in the catch

`effort` a matrix of effort

`catchability.scaling.factor`
a factor to scale the parameters

References

NONE

Examples

```
# This likelihood function constrains the selectivity on the last 2 age-groups to 1
# it doesn't help much to estimate parameters
# Simulate data
set.seed(3)
max.age <- 9
nb.of.cohort <- 50
population <- GenerateData2(max.age = max.age, nb.of.cohort = nb.of.cohort) # Generate catch using gear selec

# sample a fix number of fish each years
n.sample.per.year <- 2e3
nb.at.age.sample <- draw.sample(population$catch, sample.size = n.sample.per.year * (nb.of.cohort + 1 - max.a

# Estimate parameters, fixing selectivity for the last 2 age-groups to 1
result3 <- optim(par = c(0.2,1, rep(1e-12,max.age-2)), fn = llfunc5, catch = nb.at.age.sample, effort = popul
     lower = c(5e-2,5e-2, rep(1e-12, max.age-2)), upper = c(10,0.5, rep(1, max.age-2)), hessian = TRUE, contr
errors3 <- sqrt(abs(diag(solve(result3$hessian))))

print(cbind("Estimate" = result3$par, "Error" = errors3))
```

llfunc7 *log-likelihood function of catch at age matrix to estimate catchability, selectivity [assumed logistic] and natural mortality*

Description

This function calculate the log-likelihood of the survival model assuming fishing mortality the outer product of catchability times effort and selectivity and natural mortality is constant.

Usage

```
llfunc7(par, catch, effort, catchability.scaling.factor)
```

Arguments

`par`	a vector of four parameters: catchability, natural mortality and 2 parameters of the logistic function for gear selectivity
`catch`	a matrix containing number at age in the catch
`effort`	a matrix of effort
`catchability.scaling.factor`	a factor to scale the parameters

References

NONE

Examples

```
max.age <- 9
nb.of.cohort <- 30

population <- GenerateData3(max.age = max.age, nb.of.cohort = nb.of.cohort, verbose = TRUE)

################################################################################
# Simulate sampling
################################################################################

# sample a fix number of fish each years
n.sample.per.year <- 2e3
nb.at.age.sample <- draw.sample(population$catch, sample.size = n.sample.per.year * (nb.of.cohort + 1 - max.a
# Estimate assuming you know selectivity
lower.bound <- c(5e-2,1e-2,1,1);upper.bound <- c(15,1,20,20)

csf <- 1e-4 # catchability scaling factor

result <- optim(par = c(0.2,0.5, 10, 2), fn = llfunc7, catch = nb.at.age.sample, effort = population$effort, c
       lower = lower.bound, upper = upper.bound, hessian = TRUE)
errors <- sqrt(diag(solve(result$hessian)))

res <- cbind("Estimate" = result$par, "Errors" = errors);
dimnames(res)[[1]] <- c("Est. catchability", "Est. natural mort.", "Est. logistic par a", "Est. logistic par b
print(res)
```

```
# Calculate probability of being caught
p <- prob.for.llfunc7(result$par, population$catch, population$effort, catchability.scaling.factor = csf)

# An estimate of recruitment
rec <- rowSums(Caaa2Coaa(population$catch), na.rm = TRUE) / rowSums(p, na.rm = TRUE)
ind.rec <- Caaa2Coaa(population$catch) / p
var.ind.rec <- 1/ncol(ind.rec) * rowSums((ind.rec - outer(rec, rep(1, ncol(ind.rec))))^2, na.rm = TRUE)

par(mfrow=c(1,2))
plot(population$Rec[9:30], rec[9:30]); abline(0,1)
segments(population$Rec[9:30], rec[9:30] + sqrt(var.ind.rec[9:30]), population$Rec[9:30], rec[9:30] - sqrt(v

plot(1:22, population$Rec[9:30], type = "b", ylim = c(0.9 * min(c(rec[9:30], population$Rec[9:30])), 1.1 * ma
points(1:22, rec[9:30], type = "b", pch = 19)
```

`logistic` *logistic function*

Description

logistic function

Usage

```
logistic(a,b,x)
```

Arguments

a	1st parameter of the function
b	2nd parameter of the function
x	depend variable

References

NONE

`my.cumsum` *a cumsum that omit NA*

Description

A function to sum cumulatively rows of a matrix, replacing NA by 0.

Usage

```
my.cumsum(mat)
```

Arguments

mat	a matrix

References

NONE

`total.over.lines` *calculate cumulative sum over the lines of a matrix, ignoring NAs*

Description

Calculate the cumulative sum

Usage

```
total.over.lines(mat)
```

Arguments

`mat`	a matrix

References

NONE

`which.cohort` *A function that counts and numbers cohorts given a catch at age matrix*

Description

Number the cohort in a matrix

Usage

```
which.cohort(mat)
```

Arguments

`mat`	a matrix of data

Examples

```
mat <- matrix(NA, nrow = 5, ncol = 7)
which.cohort(mat)
```

Index

∗Topic **misc**
Caaa2Coaa, 2
Coaa2Caaa, 2
EstimateMandQ, 3
EstimateRecruitment, 4
EstimateZ, 5
GenerateData2, 6
llfunc1, 6
llfunc2, 7
llfunc3, 9
llfunc4, 11
llfunc5, 12
llfunc7, 13
logistic, 14
my.cumsum, 14
total.over.lines, 15
which.cohort, 15

Caaa2Coaa, 2
Coaa2Caaa, 2

EstimateMandQ, 3
EstimateRecruitment, 4
EstimateZ, 5

GenerateData2, 6

llfunc1, 6
llfunc2, 7
llfunc3, 9
llfunc4, 10
llfunc5, 12
llfunc7, 13
logistic, 14

my.cumsum, 14

total.over.lines, 15

which.cohort, 15