

An expert elicitation of the effects of low salinity water exposure on bottlenose dolphins

Supplementary Materials: Extended results

Cormac Booth and Len Thomas

2nd December 2020

Introduction

This document forms supplementary materials to the manuscript “An expert elicitation of the effects of low salinity water exposure on bottlenose dolphins” by Cormac Booth and Len Thomas. We describe in more detail the distributions elicited in the expert elicitation, and provide R code to recreate the resulting dose-response functions.

Distributions were elicited for three scenarios and we give results from each in turn.

Scenario 1A

Extended low salinity event in a population where animals are likely to be in a compromised health state.

The scenario was as follows. A bay, sound and estuary (BSE) environment (i.e. mean 15-25ppt) is flooded with fresh or low salinity water until salinity drops (at approx. 0.5ppt/day) to below 5ppt for an extended period. This is an environment in which animals are exposed to other significant stressors and are more likely to be in a “compromised health state”.

The question was: “for the scenario defined (above), what is the length (in days) of continuous exposure to salinity below 5 ppt, that the average BSE bottlenose dolphin in the population would need to experience to result in death (within 12 months of the start of the event)?”

The elicited distribution for this parameter (denoted d_{max} in the manuscript) was a shifted scaled beta distribution with parameters as follows:

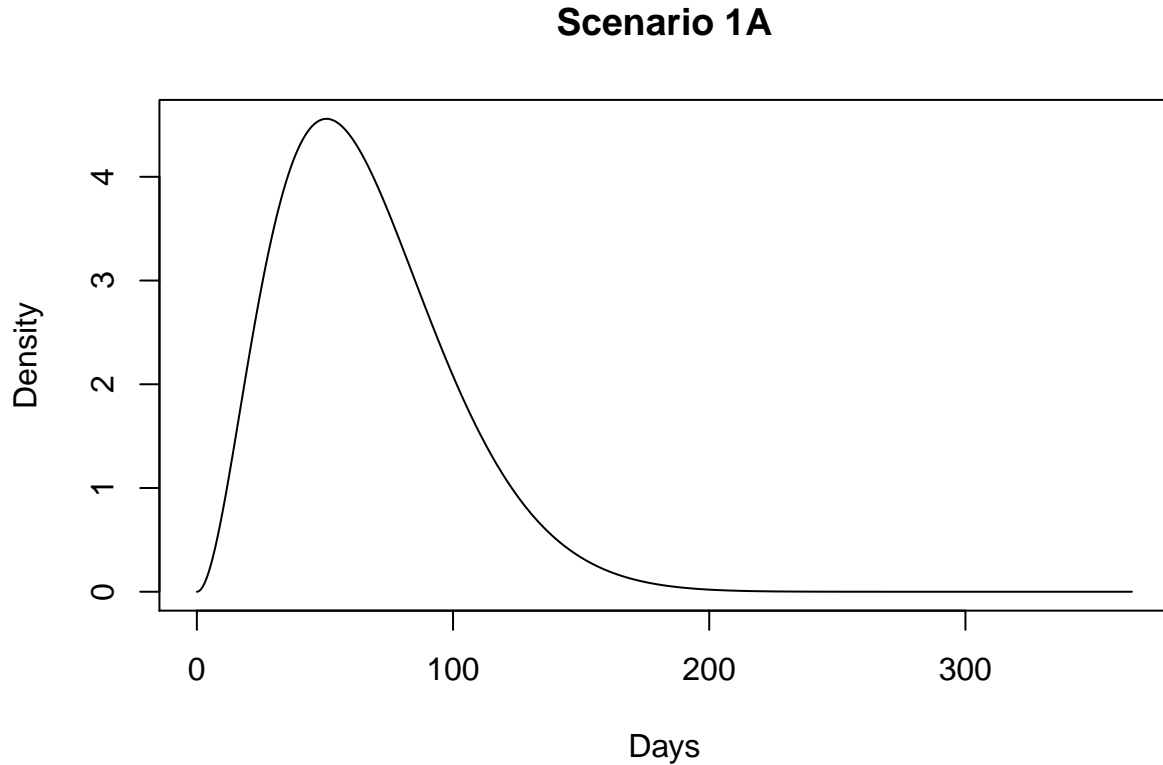
```
s.lower <- 0 #Lower bound
s.upper <- 365 #Upper bound
a <- 3.06 #first parameter (a) of beta distribution
b <- 13.8 #second parameter (b) of beta distribution
```

To demonstrate the use of these parameters, we plot the distribution:

```
d.ss.beta <- function(x, lower, upper, a, b) {
  x <- (x - lower) / (upper - lower)
  return(dbeta(x, a, b))
}

length.x <- 1000
```

```
x<-seq(0, 365, length = length.x)
d.S1 <- d.ss.beta(x, s.lower, s.upper, a, b)
plot(x, d.S1, type = "l", xlab = "Days", ylab = "Density", main = "Scenario 1A")
```



To construct a dose-response function, we also need to know the shape of the function. This was elicited separately, as detailed in the manuscript. Experts were asked “What form does the relationship between survival and length of exposure to low salinity take?” and were given a graphical tool to elicit the shape in the form of a truncated normal cumulative density function (see manuscript for details) – the two parameters elicited were a location parameter μ and a shape parameter ω . Unlike the previous question, no uncertainty was elicited from the experts on their judgments to this question, and no RIO stage was performed during the elicitation. Instead, we elicited a separate distribution from each expert together with a measure of their confidence (on a scale of 1 = least confident to 3 = most confident); experts agreed that a good consensus distribution would be obtained by sampling from their separate distributions with probability proportional to their confidence.

The elicited values were as follows:

```
res <- data.frame(mu = c(64, 71, 96, 71, 76, 96, 100),
                  sigma = c(21, 35, 34, 24, 16, 25, 25),
                  confidence = c(2, 2, 2, 2, 1, 3, 3))
res
```

```
##      mu sigma confidence
## 1   64   21           2
## 2   71   35           2
## 3   96   34           2
```

```
## 4 71 24 2
## 5 76 16 1
## 6 96 25 3
## 7 100 25 3
```

We show the resulting distribution below. Thickness of the line is proportional to the experts' confidence. We also show the confidence-weighted mean in red, and a central 50% weighted quantiles.

```
n.experts<-nrow(res)

#Function to return the value to multiply survival by
# as a function of how far you are (in percent) to
# the number of days before the average dolphin is dead
phi.multiplier <- function(perc, mu, sigma) {
  #Returns multiplier values at xvals given mu and sigma

  #Truncated normal, truncating at 0 and 100
  norm.c <- pnorm(100, mu, sigma) - pnorm(0, mu, sigma)
  if(norm.c <= 0) norm.c <- 1
  yvals <- 1 - (pnorm(perc, mu, sigma) - pnorm(0, mu, sigma)) / norm.c
  yvals[perc<0 | perc>100] <- 0
  return(yvals)
}

#Work out the survival multiplier for each expert
perc <- seq(0, 100, length = length.x)
res.mult <- matrix(NA, n.experts, length(perc))
for(i in 1:n.experts){
  res.mult[i, ] <- phi.multiplier(perc, res$mu[i], res$sigma[i])
}

#Plot results
plot(c(0,100), c(0,1), type = "n", ylab = "Survival multiplier", xlab = "Percent of days until zero survival")
for(i in 1:n.experts) {
  lines(perc, res.mult[i,], lwd = res$confidence[i])
  if(res$confidence[i] == 0)
    lines(perc, res.mult[i,], lwd = res$confidence[i], col = "blue")
}

#Work out weighted mean
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 3.6.3
```

```
## Warning: package 'lattice' was built under R version 3.6.3
```

```
## Warning: package 'survival' was built under R version 3.6.3
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

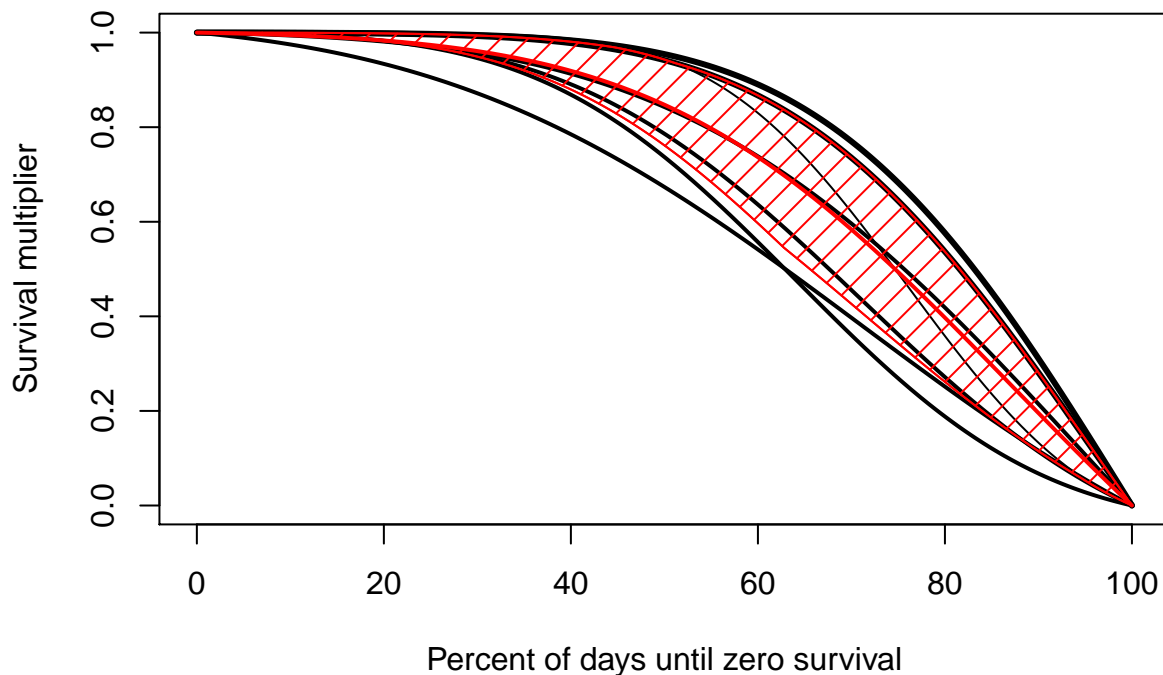
```

mean.line <- apply(res.mult, 2, wtd.mean, res$confidence)
lines(perc, mean.line, lwd = 2, col = "red")

#Weighted quantile -- note that, given the small sample size, this is approximate
lcl <- apply(res.mult, 2, wtd.quantile, weights = res$confidence, probs = 0.25)
ucl <- apply(res.mult, 2, wtd.quantile, weights = res$confidence, probs = 0.75)
lines(perc, lcl, col = "red", lwd = 1)
lines(perc, ucl, col = "red", lwd = 1)
polygon(c(perc, perc[length(perc):1]), c(ucl, lcl[length(perc):1]), col = "red", density = 10)

```

Shape of survival response



We now put these two together, sampling from number of days distribution, and then selecting from the shapes elicited with probability proportional to the experts' confidence.

We generate 10,000 of them and save to a file. The top plot shows the first 100 generated, so as to demonstrate the range of shapes produced; the bottom plot shows quantiles of the distribution generated by all 10,000; the red solid line is the median and the dashed lines are the 5th and 95th quantiles (so form a 90% interval).

```

r.ss.beta <- function(n, lower, upper, a, b) {
  x <- rbeta(n, a, b)
  x <- x * (upper - lower) + lower
  return(x)
}

```

```

generate.plot.save.realizations <- function(scenario.num, a, b, l, u, shape){
  #Purpose: function to generate realizations from the fitted survival function, save them to file, and p
  #Inputs:

```

```

# scenario.num - scenario number - used in outputs
# a, b, l, u - a, b, lower and upper parameters from shifted scaled beta distribution
# shape - data frame of expert opinions on shape of response

K <- 10000
days <- 0:365
n.days <- length(days)
scenario <- matrix(0, K, n.days)
max.days <- r.ss.beta(K, l, u, a, b)
n.experts <- nrow(shape)
experts <- sample.int(n.experts, K, replace = TRUE, prob = res$confidence)
for(i in 1:K){
  scenario[i, ] <- phi.multiplier(days/max.days[i] * 100, shape$mu[experts[i]], shape$sigma[experts[i]])
}

#Save to csv
write.csv(scenario, file = paste0("Scenario", scenario.num, ".csv"))

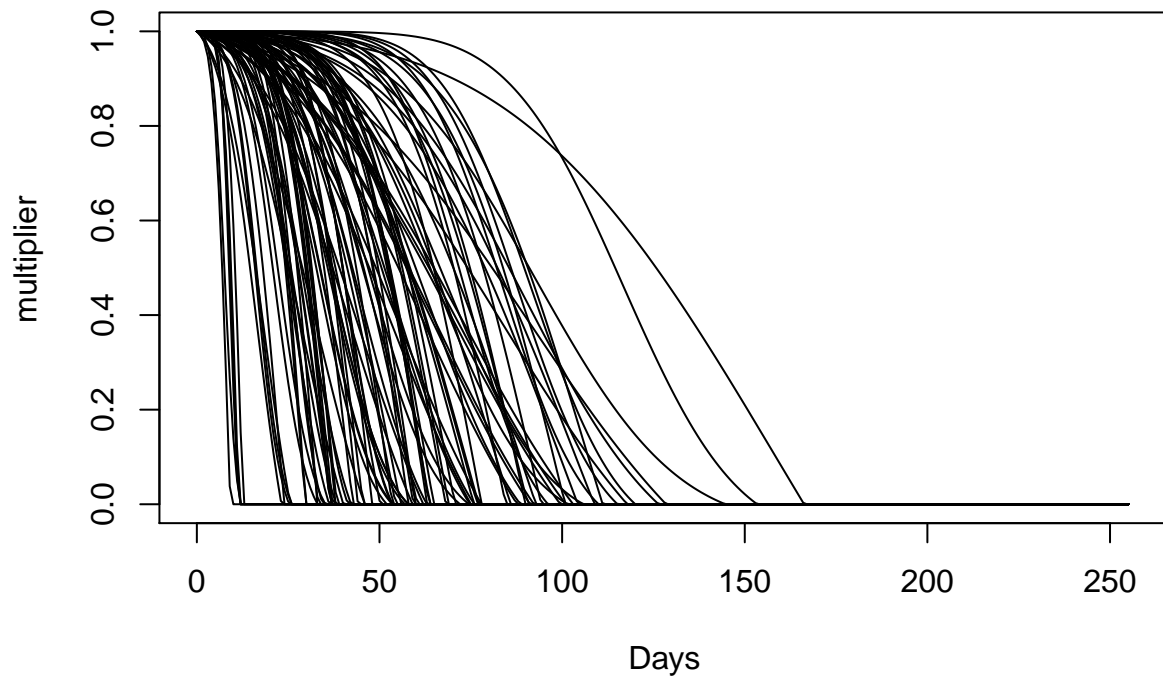
#Plot the first 100 realizations
xmax <- ceiling(max(max.days))
plot(c(0,xmax), c(0,1), type = "n", ylab = "Survival
multiplier", xlab = "Days", main = paste("100 realizations of survival response for scenario", scenario.num))
for(i in 1:100){
  lines(days[1:xmax], scenario[i, 1:xmax])
}

#Show a quantile plot
length.p <- 99
p <- seq(0.01, 0.99, length = length.p)
res.mult.quantiles <- matrix(0, length.p, xmax)
for(i in 1:xmax){
  res.mult.quantiles[, i] <- quantile(scenario[, i], probs = p)
}
plot(c(0,xmax), c(0,1), type = "n", ylab = "Survival multiplier", xlab = "Days", main = paste("Quantile plot of survival response for scenario", scenario.num))
for(i in 1:(length.p/2)){
  polygon(c(1:xmax, xmax:1), c(res.mult.quantiles[i, ], res.mult.quantiles[length.p-i+1, xmax:1]), col = "red", lty = 2)
}
#plot median and 90iles
lines(1:xmax, res.mult.quantiles[50, ], col = "red")
lines(1:xmax, res.mult.quantiles[5, ], col = "red", lty = 2)
lines(1:xmax, res.mult.quantiles[95, ], col = "red", lty = 2)
invisible(NULL)
}

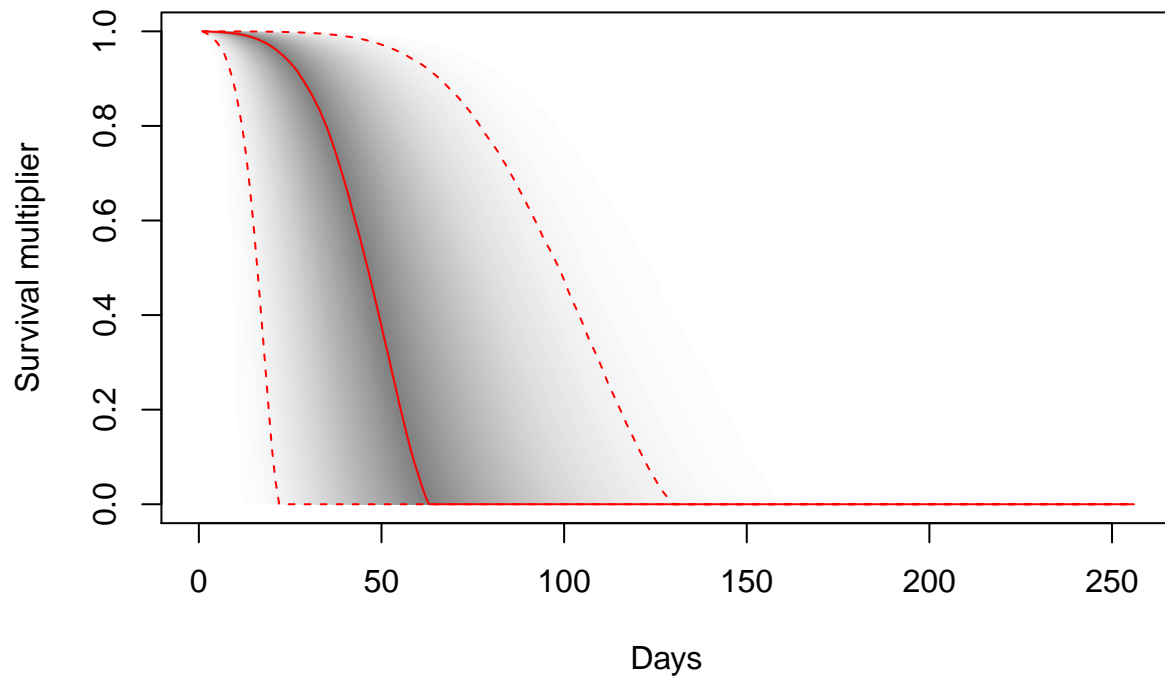
#Set seed so the realizations are reproduceable
set.seed(31823632)
generate.plot.save.realizations ("1A", a, b, s.lower, s.upper, res)

```

100 realizations of survival response for scenario 1A



Quantiles of survival response for scenario 1A



Scenario 1B

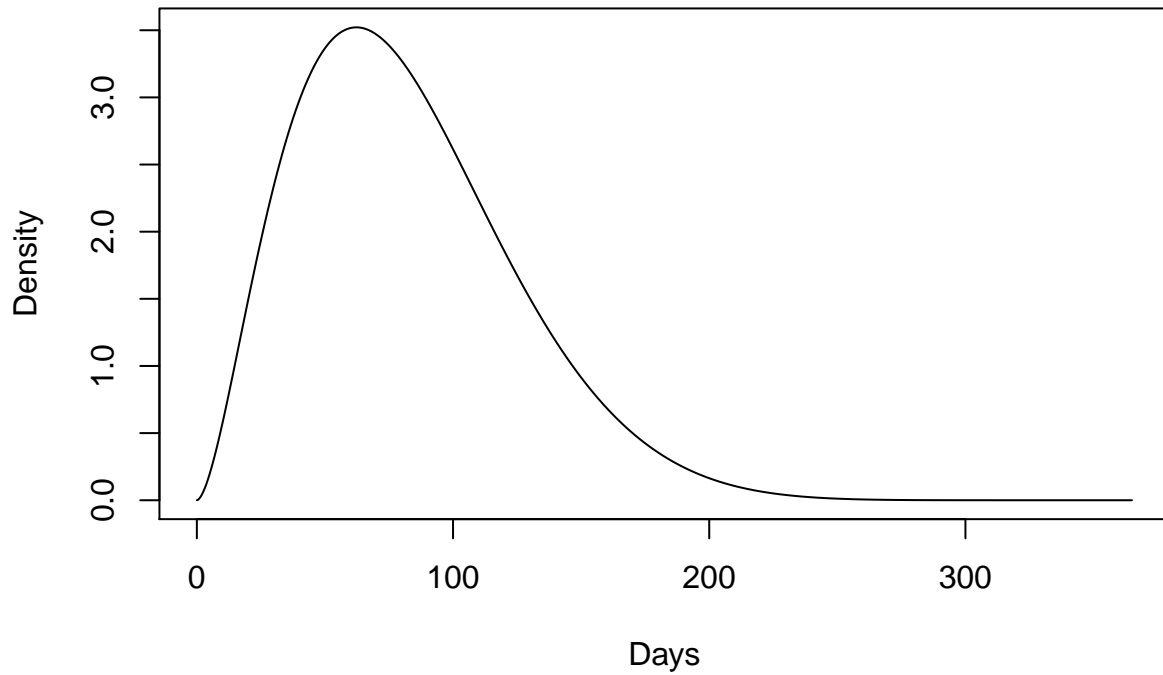
Extended low salinity event in a population where animals are likely to be in a healthy state.

We repeat the same approach for this scenario.

```
s.lower <- 0
s.upper <- 365
a <- 2.72
b <- 9.36

length.x <- 1000
x<-seq(0, 365, length = length.x)
d.S1 <- d.ss.beta(x, s.lower, s.upper, a, b)
plot(x, d.S1, type = "l", xlab = "Days", ylab = "Density", main = "Scenario 1B")
```

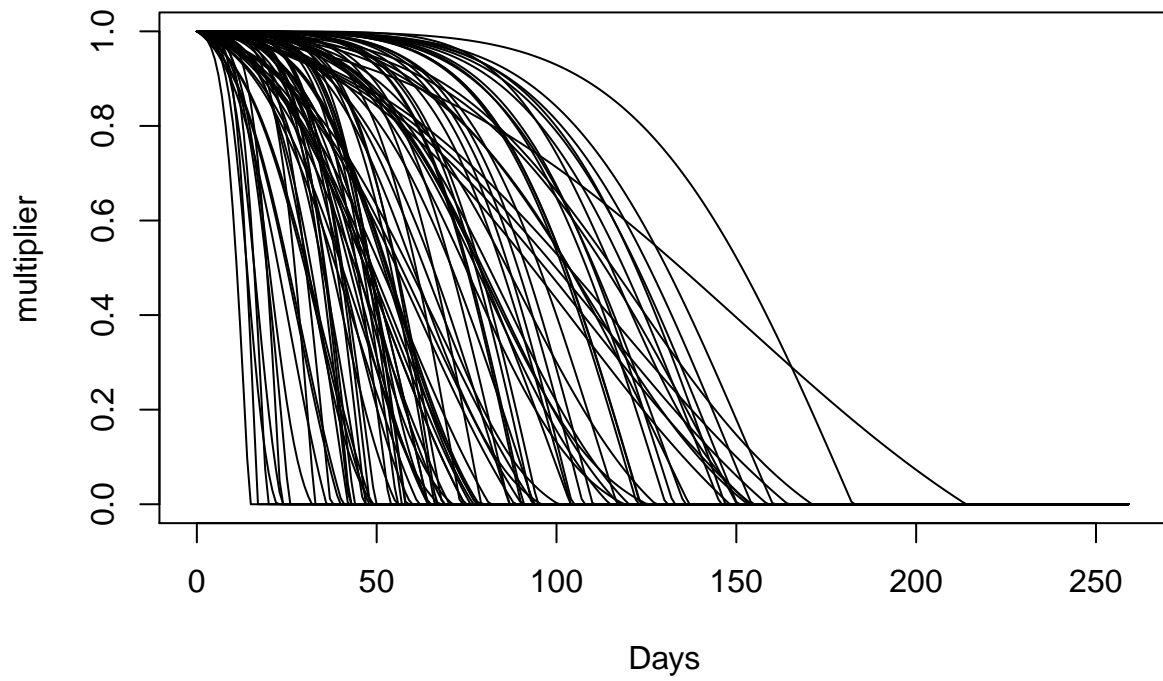
Scenario 1B



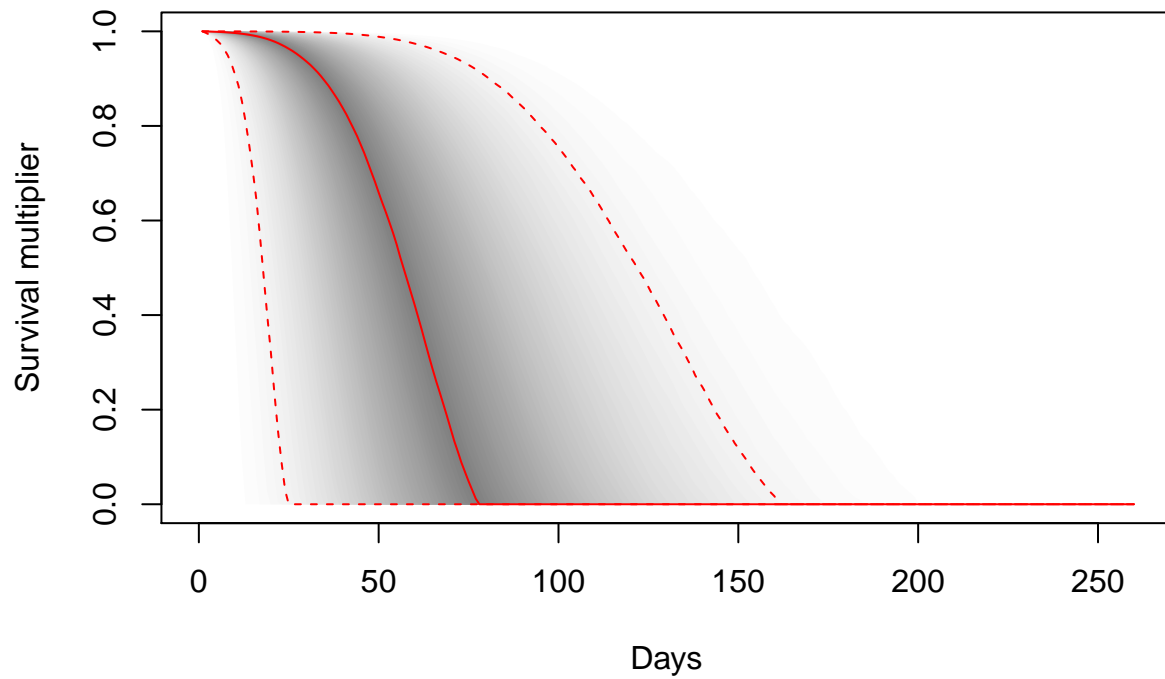
The same dose-response shape distribution was used for all three scenarios. Hence, putting the above distribution for d_{max} with the previously shown dose-response shape yields the following dose-response functions (again 10,000 samples were generated).

```
#Set seed so the realizations are reproduceable  
set.seed(8432234)  
generate.plot.save.realizations ("1B", a, b, s.lower, s.upper, res)
```


100 realizations of survival response for scenario 1B



Quantiles of survival response for scenario 1B



Scenario 2

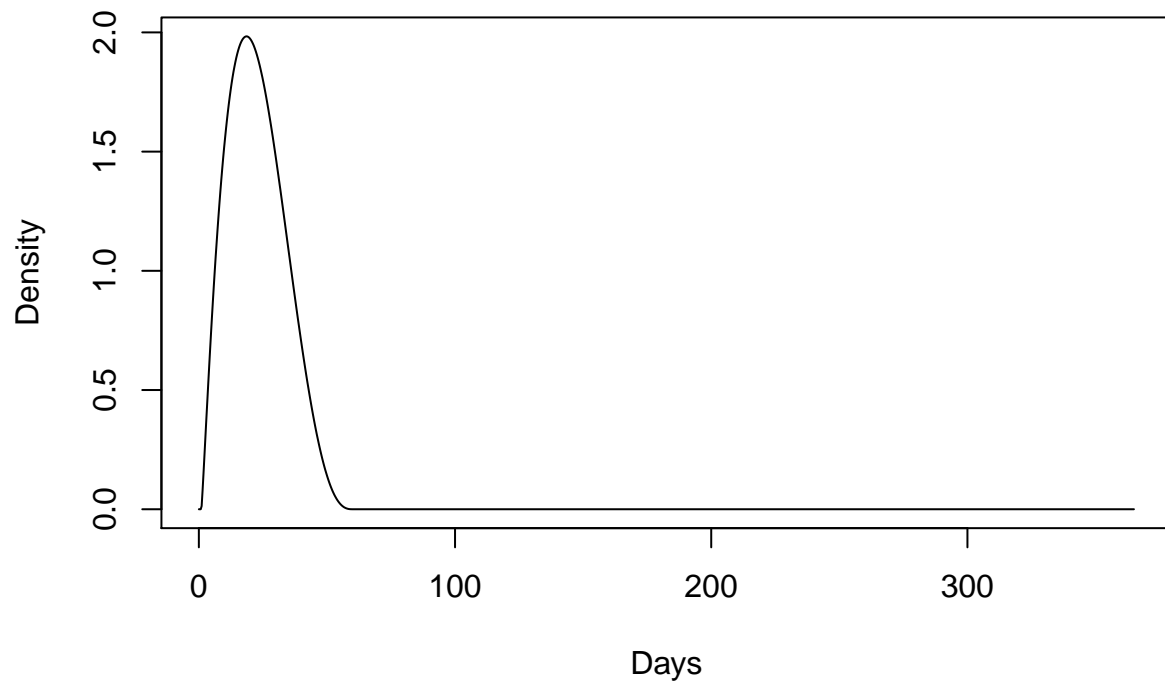
Acute salinity change event

Lastly, we use the same approach for scenario 2.

```
s.lower <- 1
s.upper <- 60
a <- 2.14
b <- 3.68

length.x <- 1000
x<-seq(0, 365, length = length.x)
d.S1 <- d.ss.beta(x, s.lower, s.upper, a, b)
plot(x, d.S1, type = "l", xlab = "Days", ylab = "Density", main = "Scenario 2")
```

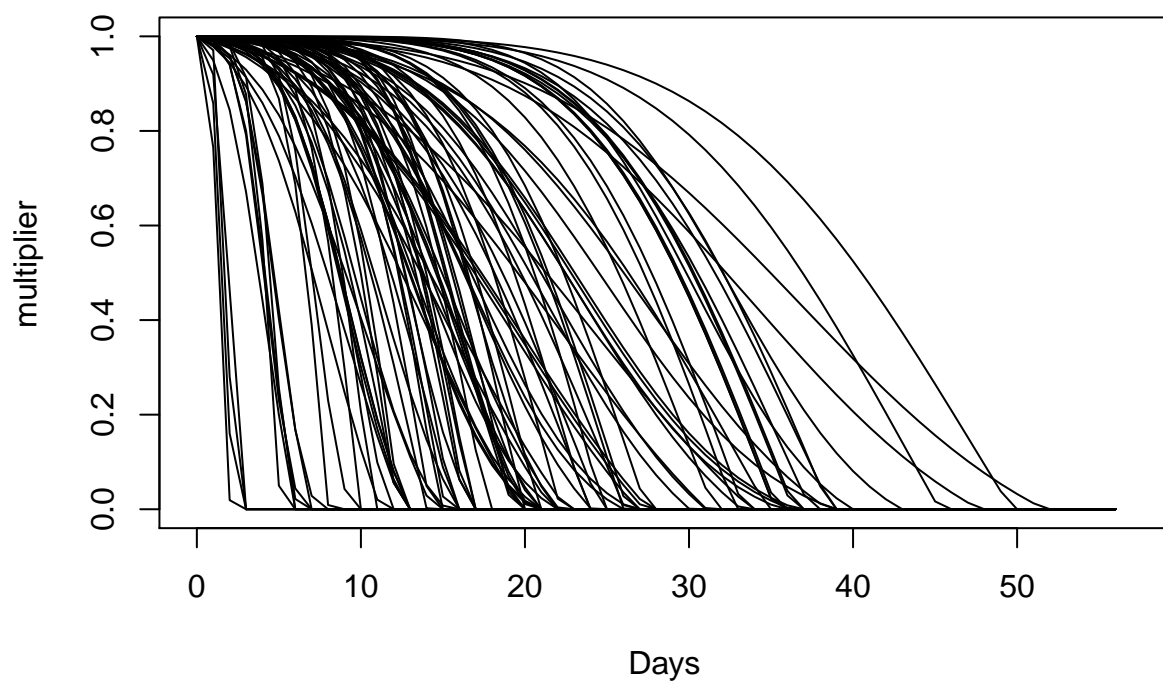
Scenario 2



Again, putting the above distribution for d_{max} with the previously shown dose-response shape yields the following dose-response functions (10,000 samples were generated).

```
#Set seed so the realizations are reproduceable  
set.seed(2867)  
generate.plot.save.realizations (3, a, b, s.lower, s.upper, res)
```

100 realizations of survival response for scenario 3



Quantiles of survival response for scenario 3

