

BRISTOL UNIVERSITY

MATHEMATICS

---

**Theory of Inference Project**

**MATH 35600**

---

OWEN WAGHORN - 1807507

BAYLEE DEVERS - 1812473

ALFRED CHEUNG - 1847559

November 20, 2021

# 1 Introduction

In this practical, we are considering an experiment where toads *uglibus* frogs have been placed in an enclosed pond. 28 different sizes of frogs are used and measured in centimeters, 10 of each size of frog are placed in the pond, and the number of frogs in each size category that are killed by predators within a 3 day time period is recorded. The data we use for this practical can be found at <https://mfasiolo.github.io/TOI/frogs.RData>.

We consider two different models for the probability of a frog being killed. The first model is a modified logistic function, given as:

$$P_L(s) = \frac{e^{\epsilon(\phi-s)}}{1 + e^{\beta\epsilon(\phi-s)}} \quad (1)$$

where  $s$  is the size of the frog,  $\phi$  is a location parameter such that  $\mathbb{P}(s = \phi) = 0.5$ ,  $\epsilon$  controls the rate of change of  $\mathbb{P}(s)$  with  $s$ , and  $\beta$  controls the asymmetry of the function.

The second model is the generalised Ricker model:

$$P_R(s) = b \left\{ \frac{s}{a} \exp\left(1 - \frac{s}{a}\right) \right\}^\alpha \quad (2)$$

where  $a$ ,  $b$ , and  $\alpha$  are model parameters.

We assume that the number of frogs killed in the  $i$ -th experiment,  $kill_i$ , follows a binomial distribution:

$$kill_i \sim \text{binom}(p_i, n = 10) \quad (3)$$

where  $p_i = P(s_i)$  is modelled by  $P_L(s_i)$  or  $P_R(s_i)$

# 2 Comparison of Models

To begin comparing the models, we minimise the negative log-likelihood of the data under both models using *optim* in R. We then plot these estimated values against the observed values from the data. We used initial values of the model parameters given to us that are good for optimising the model. For model (1) we use  $\epsilon = -0.25$ ,  $\beta = 5$  and  $\phi = 13$ . For model (2), we used  $a = 8$ ,  $b = 0.25$ ,  $\alpha = 3$ .

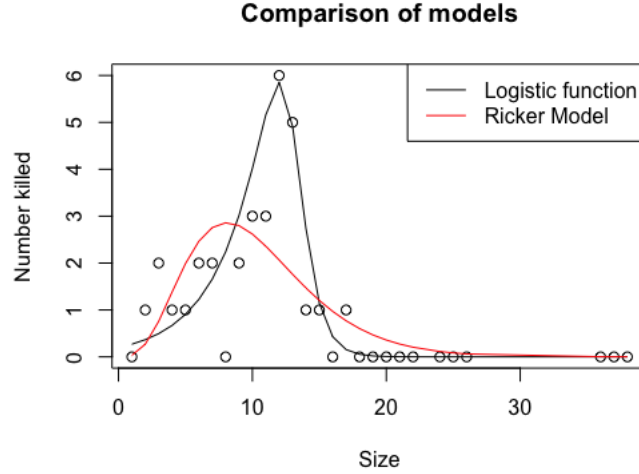


Figure 1: Plot of estimated values against observed

What we observed from figure 1 is that the logistic function seems to fit the data better than the Ricker model (note that the dots are the observed values). To continue the comparison, we computed the Akaike's information criterion (AIC) for each of the models. The AIC is an estimator of prediction error and thereby relative quality of statistical models for a given set of data. The model with the minimum AIC suggests the best fitted model. The computed values of the AIC for the Logistic and Ricker model were 52.68360 and 65.85527 respectively. This seems to support our hypothesis, that the Logistic function models the data better. The difference between the AIC values is not massive though, so the Ricker model may still offer some utility.

### 3 Argument on Probability of Predation

Some experts on frogs behaviour suggested that the probability of predation should monotonically increase with size since frogs with larger sizes find it more difficult to hide. Under the assumption that  $\epsilon < 0$ , this will happen when  $\beta = 1$ . The estimated model parameters for the logistic function are  $(\hat{\epsilon}, \hat{\beta}, \hat{\phi}) = (-0.3019021, 4.408179, 12.95549)$ . These estimates were the result of the **optim** function in R.

Our estimate of  $\beta$  doesn't seem to be close to 1. To further analyse this, we used a confidence interval to assess whether the value  $\beta = 1$  is compatible with the

data. With an initial value of  $\beta = 5$  given, our estimated value for  $\beta$  is given as  $\hat{\beta} = 4.4081788$ . Given this estimate, the 95% confidence interval is  $[1.8377483, 6.9786093]$  which supports our belief that  $\beta = 1$  is not compatible with the data. Another reason we believe this  $\beta$  is not compatible with the data is due to the setup of the logistic function. We see in figure 1 that the logistic function peaks at a value then has a tail at either end. Looking at the limit of the logistic function, we find that when  $\epsilon < 0$  and  $\beta = 1$

$$\lim_{x \rightarrow \infty} \frac{1 + e^{\epsilon(\phi - x)}}{e^{\epsilon(\phi - x)}} = 1$$

If this was true, we would see that as the size increased, the amount of frogs getting killed would increase, which is not the case in the data. Also, after a certain point, we would have that  $P_L(s) = 1$  which would see every single frog after the point dying, which is also untrue as after approximately size 15, the amount of frogs getting killed decreases dramatically. We will show this explicitly by simulating values using  $(\hat{\epsilon}, 1, \hat{\phi})$ , we graph these simulated values with the actual data.

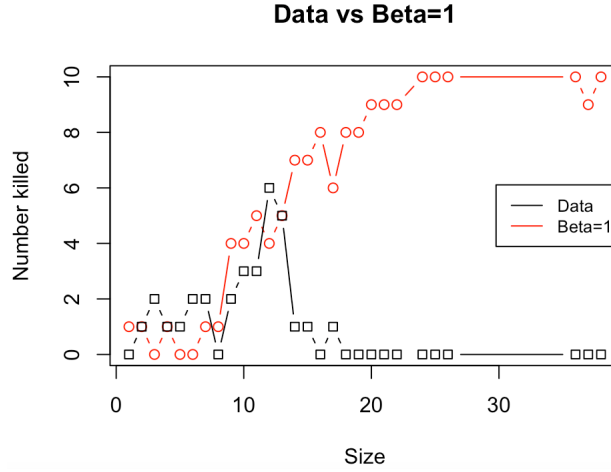


Figure 2: Comparison of data with a beta=1 sample

These simulated values confirm our previous statements about the model with  $\beta = 1$ . We see that it models the data reasonably well until around size 15, where the limit is reached and the model will say almost all 10 frogs are killed every time when in fact we see that the opposite is true from the data. We conclude that that the probability of predation does not increase monotonically with size and that  $\beta = 1$  is not compatible with the data given.

## 4 Endangered Species

We have been asked to provide guidance to conservationists about which size of frogs in the wild to focus on protecting. Based on the data seen in figure 1, we see that frogs between the size of 10-15cm are dying more than other sizes, so we have reason to believe that these have a higher probability of being killed. This section will cover our analysis on the size  $s^*$  that maximises  $P_L(s)$ , allowing us to better understand which group of frogs have the highest probability of being killed. We will also produce confidence intervals for this estimation of maximal predation using the delta method.

To find the value of  $s$  we are looking for, we must find a function  $g(\theta)$  such that  $s^* = g(\theta)$ . Here we have that  $\theta = \{\epsilon, \beta, \phi\}$ . Through rearrangement of (1), we find that the function of  $g$  is

$$s^* = g(\theta) = \phi - \log(y) + \log(e^\epsilon - ye^{\epsilon\beta}) \quad (4)$$

where  $y$  is the maximum probability of getting killed from the logistic model seen in figure 1. To find our estimate for the size when predation is maximal  $s^*$ , we simply compute  $\hat{s}^* = g(\hat{\theta})$  where  $\hat{\theta}$  are the estimated values for  $\epsilon, \beta$  and  $\phi$  calculated in section 2. Hence, we can find that the size that maximises probability of getting killed  $P_L(s)$  is 12.95453, therefore, we suggest that the conservationists should focus their efforts on frogs with a size of approximately 13cm.

## 5 Confidence interval of estimation $\hat{s}^*$

To test the reliability of our estimate for the value of  $s^*$ , we will find the confidence interval of our estimation  $\hat{s}^*$ . Although  $g(\theta)$  is non-linear, we believe we can assume the linearity of  $g(\theta)$  and carry out the delta method, without losing much information. Below is our linear approximation of  $g(\theta)$ ,

$$\hat{s}^* = g(\hat{\theta}) \approx g(\theta) + \nabla g(\theta)^T (\hat{\theta} - \theta) \quad (5)$$

where  $\nabla g(\theta)$  is the gradient of  $g$ .

Note that we can get a vector of the gradient of  $g$  by differentiating equation (4), which then gives

$$\nabla g(\theta) = \nabla g(\epsilon, \beta, \phi) = \left( \frac{e^\epsilon - y\beta e^{\beta\epsilon}}{e^\epsilon - ye^{\beta\epsilon}}, \frac{y\epsilon e^{\beta\epsilon}}{ye^{\beta\epsilon} - e^\epsilon}, 1 \right) \quad (6)$$

By using the gradient of  $g$  calculated from (6) and the hessian matrix of the minimised negative log likelihood of model (1) as the estimation of  $cov(\hat{\theta})$ , we can use the delta method to find the variance of  $\hat{s}^*$

$$var(\hat{s}^*) = \nabla g(\theta)^T cov(\hat{\theta} - \theta) \nabla g(\theta) \quad (7)$$

We set a significance level at 5% and derive that the confidence interval for  $s^*$  is [5.867233, 19.9789]. Unfortunately, the confidence interval is very large, which means that the estimate is difficult to trust. We believe this is due to the small sample of  $n=10$  of each frog size giving high values in  $cov(\hat{\theta})$ .

$$cov(\hat{\theta}) = \begin{bmatrix} 778.927384 & -13.474102 & -7.881211 \\ -13.474102 & 1.486754 & -2.095862 \\ -7.881211 & -2.095862 & 7.491799 \end{bmatrix}$$

This causes the variance of  $s^*$  to be very large. A larger sample, perhaps a sample size of at least  $n=100$  would improve the accuracy and achieve a more accurate approximation.

## 6 R code

```
load(file = url("https://mfasiolo.github.io/TOI/frogs.RData"))
head(frogs)

#We first define the modified logistic function and the Ricker model in R.
#Note that theta contains all of the model parameters.
#For the logistic model, that is ,epsilon,beta and phi.
#For in the Ricker model, that is a,b and alpha.

#Logistic function
P_l <- function(s,theta){
  e <-theta[1]
  beta <- theta[2]
  phi <-theta[3]
  p<-(exp(e*(phi - s)))/(1 + exp(beta*e*(phi - s)))
  return(p)}

#Ricker model
P_r <- function(s,theta){
  a <- theta[1]
  b <- theta[2]
  alpha <- theta[3]
  p <- b*(((s/a)*exp(1 - s/a))^alpha)
  return(p)}

#Here we order the data based on the size of the frogs to make it
#easier to plot
frogs <-as.data.frame(frogs)
frogs <- frogs[order(frogs["size"]),]
x <- frogs$size
y <- frogs$killed

#The two following functions compute the negative log-likelihood of
#each model using the binomial model given.
P_l_log<-function(s,y, theta){
  p <- P_l(s, theta)
  loglik <- sum(dbinom(y,10,p, log=TRUE))
  -loglik}

P_r_log<-function(s,y,theta){
  p <- pr(s, theta)
  loglik <- sum(dbinom(y,10,p, log=TRUE))
  -loglik}
```

```

#The code below minimises the negative log-likelihood under each model
#using optim. Once we have the minimised values, we plot them both against
#the observed values to compare the accuracy of each model

beta0 <- 5 #Initial values given
e0 <- -0.25
phi0 <-13
theta01 <- c(e0, beta0, phi0) #Putting the values into a list as stated earlier
qfit1 <- optim(theta01 ,P_l_log,method="BFGS",hessian= TRUE,s=x,y=y)
qfit1
print(pl(qfit1$par[2],qfit1$par))

a0 <- 8
b0 <- 0.25
alpha0 <- 3
theta02 <- c(a0, b0, alpha0)
qfit2 <- optim(theta02 ,P_r_log,method="BFGS",hessian= TRUE,s=x,y=y)
qfit2

plot(x,y,xlab="Size",ylab="Number_killed",main="Comparison_of_models")
lines(x,10*pl(x,qfit1$par))
lines(x,10*pr(x,qfit2$par),col="red")
legend("topright", 95, legend=c("Logistic_Function", "Ricker_Model"),
      col=c("black", "red"), lty=1:1, cex=0.8)

#Here we compute the AIC of each model to see which model
minimises it better
AIC1 <- 2*P_l_log(x,y,qfit1$par)+length(qfit1$par)
AIC2 <- 2*P_r_log(x,y,qfit2$par)+length(qfit2$par)
Comp<-c(AIC1,AIC2)
Comp

#Here we produce confidence intervals for each of the model parameters
theta1.sd <- diag(solve(qfit1$hessian))^0.5
ci1 <- c(qfit1$par - 1.96*theta1.sd , qfit1$par + 1.96*theta1.sd)
ci1
conf1<-c(ci1[1],ci1[4],ci1[2],ci1[5],ci1[3],ci1[6])
conf1
theta2.sd <- diag(solve(qfit2$hessian))^0.5
ci2 <- c(qfit1$par - 1.96*theta1.sd , qfit1$par + 1.96*theta1.sd)
ci2
conf2<-c(ci2[1],ci2[4],ci2[2],ci2[5],ci2[3],ci2[6])
conf2

```



```

#This function creates a sample of the model for a given theta
Samp<-function(theta){
  vals=c()
  for (i in 1:28){
    val<-P_l(i,theta)
    numb<-rbinom(1,10,val)
    vals[i]=numb
  }
  return(vals)
}

#We create a plot of when beta=1 with our estimated values and compare
#to the actual data
thetatest<-c(qfit1$par[1], 1, qfit1$par[3])
plot(x,Samp(thetatest),type="b",col="red",xlab="Size",ylab="Number_killed",
main="Data_vs_Beta=1")
lines(x,y,type="b",pch=22,)
legend("right", 95, legend=c("Data", "Beta=1"),
      col=c("black", "red"), lty=1:1, cex=0.8)

#This is the inverse function which is given as equation (4) in
#the report
gfunc <- function(theta){
  y <- max(P_l(x,qfit1$par))
  e <- theta[1]
  beta <- theta[2]
  phi <- theta[3]
  s <- phi - log(y) + log(exp(e)-y*exp(e*beta)) # inverse function by hand,
  #y set as approximately 0.6 which is the probability when s = 12.
}

#This code produces the estimated s^* using the gfunc above
smax <- gfunc(qfit1$par)
smax

#Here we calculate the value of y needed for the value of g(theta)
yconst <- max(P_l(x,qfit1$par))
ehat<-qfit1$par[1] #estimated values of the model parameters
betahat <- qfit1$par[2]
phihat <- qfit1$par[3]

#This calculates the gradient of g which is given as (6) in the report
grade <- (exp(ehat)-yconst*betahat*exp(ehat*betahat))/
(exp(ehat)-yconst*exp(ehat*betahat)) # gradient by hand
gradbeta <- (yconst*ehat*exp(ehat*betahat))/

```

```

(yconst*exp(ehat*betahat)-exp(ehat))
gradphi <- 1
gradgfunc <- c(grade,gradbeta,gradphi)
#This code calculates the variance of s^* as in equation (7) in the
#report
vargfunc <- t(gradgfunc)%*%qfit1$hessian%*%gradgfunc
stderrorgfunc <- sqrt(vargfunc)

#This code produces the confidence interval for s^*
upbound <- smax + 1.96*stderrorgfunc
lowbound <- smax - 1.96*stderrorgfunc
print(upbound)
print(lowbound)

```