

Individual Assignment

# Optimization Techniques

---

Inderpreet Singh RANA

20th February 2022

## Index

- Project Overview
- Method
- Implementation
- Conclusion
- References

## Project Overview:

Maximizing the likelihood of a binomial distribution by optimizing the likelihood of a realized collection of data.

**Binomial Distribution:** A frequency distribution of the possible number of successful outcomes in each number of trials in each of which there is the same probability of success.

**Maximum likelihood estimation (MLE)** is a method of estimating the parameters of an assumed probability distribution, given some observed data. This is achieved by maximizing a likelihood function so that, under the assumed statistical model, the observed data is most probable.

**Data used:** Data on reed frog predation experiments.

Data loaded using “emdbook” library in R

**Data description:** Data on lab experiments on the density- and size-dependent predation rate of an African reed frog from Vonesh and Bolker 2005.

### Variables:

- **Density:** initial tadpole density (number of tadpoles in a 1.2 x 0.8 x 0.4 m tank)
- **Pred:** factor: predators present or absent
- **Size:** factor: big or small tadpoles
- **Surv:** number surviving
- **Propsurv:** proportion surviving (=surv/density)
- **TBL:** tadpole body length in mm
- **Kill:** number killed out of 10, in 3 days
- **Initial:** initial number/density (300 L tank)
- **Killed:** number killed by 3 dragonfly larvae in 14 days

*\*\*Note: Because predation on tadpoles is size and density dependent, I did a subset on this data to a single class of small and density of 10 for all treatments including a predator.*

## Method:

For the given dataset, each individual or the per-trial probability of being eaten by a predator is a binomial process. That is, they can survive or die during the interval.

That is the likelihood that **k** out of **N** individuals are eaten as a function of the per capita predation probability *p* is:

$$Prob(k | p, N) = \binom{N}{k} p^k (1 - p)^{N-k}$$

Since the observations are independent, the joint likelihood of the whole data set is the product of the likelihood of each individual observation. Hence, if we have **n** observations, each with the same total number of tadpoles **N**, and the number of tadpoles killed in the **i**<sup>th</sup> observation is **k<sub>i</sub>**, then the likelihood is:

$$L = \prod_{i=1}^n \binom{N}{k_i} p^{k_i} (1 - p)^{N-k_i}$$

It is assumed that the data is binomially distributed – the binomial distribution is the natural choice for data that are represented as *k* ‘successes’ out of *N* ‘trials’. We conventionally work in terms of the log-likelihood (LL), which is:

$$LL = \sum_{i=1}^n \left[ \log \binom{N}{k_i} + k_i \log(p) + (N - k_i) \log(1 - p) \right]$$

## Implementation:

Now we can implement the same equation in R as follows

```
killed <- rfp_sub$density-rfp_sub$urv
N=rfp_sub$density
p=0.5

# Expression of data likelihood (log scale)
sum(dbinom(killed, size=N, prob=p, log=TRUE))
```

Which gives us the value: -12.8038

Now, there is only one parameter in this calculation that is **p**, because we know how many total individuals that we started with ( $N = 10$  for each trial) and how many survived in each trial ( $k = 7, 5, 9$ , and  $9$ ). Based on this we now want to solve for the most likely value of **p** given our observations of **N** and **surv**.

In essence we do this by picking a possible value of **p** which can only range from 0 to 1, calculating the log-likelihood using the above equation, checking with multiple values of **p**, completing the equation, etc. until we exhaust all possible values of **p** and identify the one having the highest likelihood value.

We convert this data into binomial distribution using the `dbinom()` function in R and apply some parameters:

Given our observed **k** (number killed), and **N** = 10 for each trial, what is the likelihood that **p** = 0.5 for each of our trials

```
# Evaluate data likelihood with p=0.5
dbinom(num_killed,size=10,prob=0.5)

#[1] 0.117187500 0.246093750 0.009765625 0.009765625
```

We can see that given our data, fixed sample size, and model (with  $p = 0.5$ ), our observed outcomes are very unlikely.

Now let's check the likelihood of observing all 4 of our outcomes, i.e, the joint probability of our data?

```
# Joint data likelihood
prod(dbinom(num_killed,size=10,prob=0.5))

## [1] 2.750312e-06
```

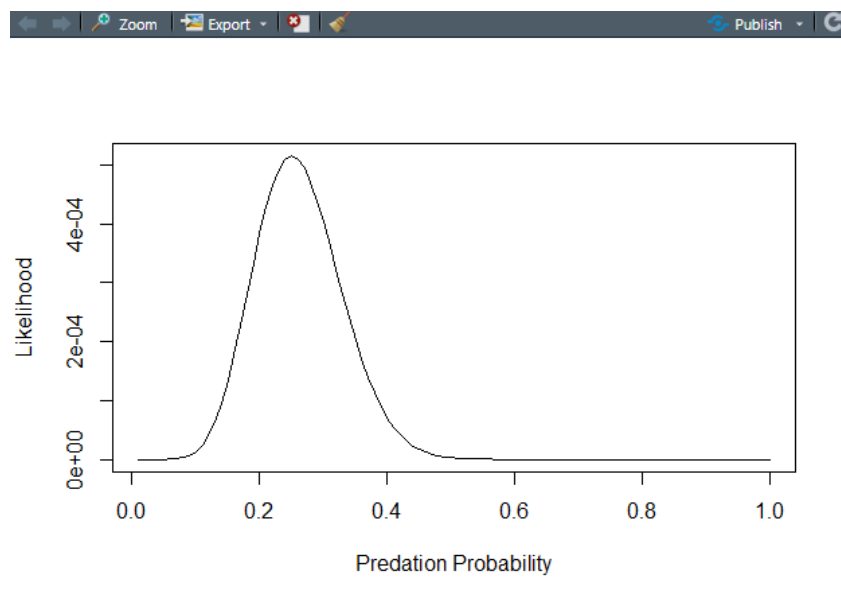
The joint likelihood values will be less than 1 and gets smaller and smaller each time we add more data. Therefore, we used log-likelihoods which yield larger numbers having better mathematical properties. Taking the log of a value  $< 1$  yields a negative number, which is why we often see that our log likelihood values are negative.

For now, we can build on this above process to estimate the likelihood function over the entire possible parameter space (probability of being eaten- which can range from 0 to 1).

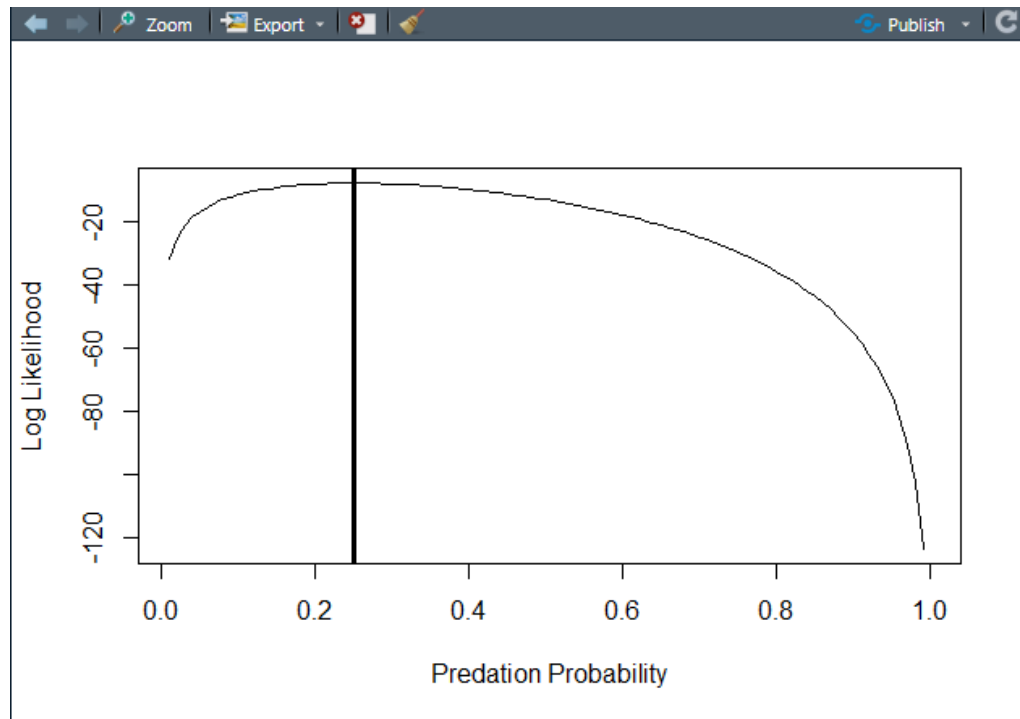
First, we make a sequence of 100 possible parameter values from 0.01 to 1 and make an empty storage vector for the likelihoods we'll calculate and then we prepare the data for visualization.

```
# prepare for visualizing the likelihood across parameter space
p <- seq(0.01, 1, length=100)
Like <- numeric(length=100)
```

We can use a loop here, for every value of  $p$  (a sequence of 100 values) we can calculate the binomial probability and store it in the *Like* vector.



Since our objective is to maximize the log-likelihood, we take the log and replot and add a line to indicate the maximum log likelihood:



We can now get the maximum likelihood in R using the `max` function which gives us the value 0.25.

we can also use the `optim()` or `mle2()` functions in R to find the maximum likelihood estimates.

Although we seek the most likely, or maximum likelihood estimate, in practice we generally minimize the negative log-likelihood. For this, first write a function to calculate the binomial negative log-likelihood function and estimate parameter  $p$ .

## Conclusion:

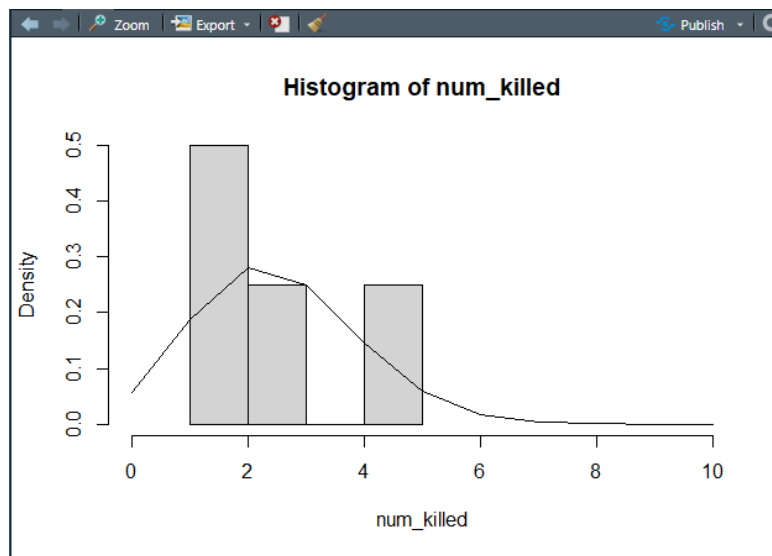
Using the `optim()` function to minimize the negative log-likelihood function given a vector of starting parameters and our data. The starting parameters can be assumed but needs to be reasonable for the function to work, given that there is only one estimable parameter **p** in the binomial function, we need only provide a starting estimate for it.

```
binomNLL1 <- function(p, k, N) {  
  -sum(dbinom(k, size=N, prob=p, log=TRUE))  
}  
  
opt_results <- optim(fn=binomNLL1, par = c(p=0.5),  
  N = 10, k = num_killed,  
  method = "BFGS")
```

From the **opt\_results** output we see that:

- Our '**p**' value or MLE is 0.2500002, which seems to be very close to the theoretical answer based on average value which in this case would be 25%
- the **value** of 7.571315 here is the negative log-likelihood for the model, when we convert it to the likelihood using `exp()` function, we get 0.0005150149.
- The convergence value of 0 means convergence has been achieved. (a value of 1 would have meant the process failed to converge).

We can now plot the observed outcomes against predictions with our maximum likelihood model:





## References:

- [https://www.researchgate.net/publication/233438268\\_Maximum\\_Likelihood\\_Estimation\\_of\\_the\\_Log-Binomial\\_Model](https://www.researchgate.net/publication/233438268_Maximum_Likelihood_Estimation_of_the_Log-Binomial_Model)
- <https://stats.stackexchange.com/questions/181035/how-to-derive-the-likelihood-function-for-binomial-distribution-for-parameter-es>
- <https://kevintshoemaker.github.io/NRES-746/LECTURE4.html>
- <https://kevintshoemaker.github.io/NRES-746/LECTURE2.html>
- <https://towardsdatascience.com/maximum-likelihood-estimation-in-r-b21f68f1eba4>
- [https://www.montana.edu/rotella/documents/502/binom\\_like.pdf](https://www.montana.edu/rotella/documents/502/binom_like.pdf)