

Bayesian statistics with R

2. The likelihood

Olivier Gimenez

March 2021

Likelihood

- Usually, when talking about probability distributions, we assume that we know the parameter values.

- Usually, when talking about probability distributions, we assume that we know the parameter values.
- In the real world, it is usually the other way around.

A question of interest might be for example:

We have observed 3 births by a female during her 10 breeding attempts. What does this tell us about the true probability of getting a successful breeding attempt from this female? For the population?

- We don't know what the probability of a birth is.
- But we can calculate the probability of getting our data for different values:

```
dbinom(x=3,size=10,prob=0.1)
```

```
#> [1] 0.05739563
```

- We don't know what the probability of a birth is.
- But we can see what the probability of getting our data would be for different values:

```
dbinom(x=3,size=10,prob=0.9)
```

```
#> [1] 8.748e-06
```

- We don't know what the probability of a birth is.
- But we can see what the probability of getting our data would be for different values:

```
dbinom(x=3,size=10,prob=0.25)
```

```
#> [1] 0.2502823
```



```
dbinom(x=3,size=10,prob=0.1)
#> [1] 0.05739563
dbinom(x=3,size=10,prob=0.9)
#> [1] 8.748e-06
dbinom(x=3,size=10,prob=0.25)
#> [1] 0.2502823
```

So we would be more likely to observe 3 births if the probability is 0.25 than 0.1 or 0.9.

The likelihood

- This reasoning is so common in statistics that it has a special name:

The likelihood

- This reasoning is so common in statistics that it has a special name:
- **The likelihood** is the probability of observing the data under a certain model.

The likelihood

- This reasoning is so common in statistics that it has a special name:
- **The likelihood** is the probability of observing the data under a certain model.
- The data are known, we usually consider the likelihood as a function of the model parameters $\theta_1, \theta_2, \dots, \theta_p$

$$L = P(\theta_1, \theta_2, \dots, \theta_p \mid \text{data})$$

Likelihood functions

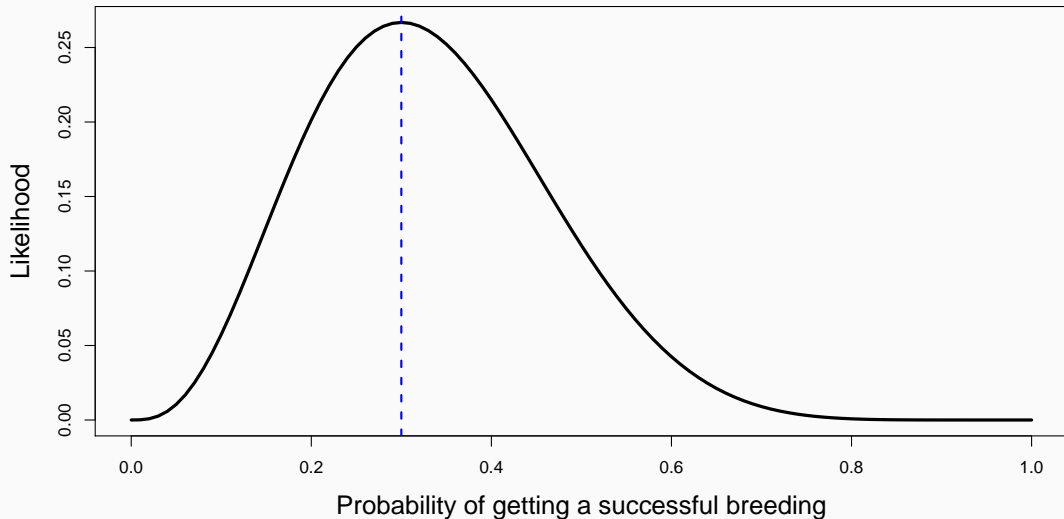
We may create a function to calculate a likelihood:

```
lik.fun <- function(parameter){  
  ll <- dbinom(x=3, size=10, prob=parameter)  
  return(ll)  
}
```

```
lik.fun(0.3)  
#> [1] 0.2668279
```

```
lik.fun(0.6)  
#> [1] 0.04246733
```

Maximize the likelihood (3 successes out of 10 attempts)



The *maximum* of the likelihood is at value 0.3

Maximum likelihood estimation

- There is always a set of parameters that gives you the highest likelihood of observing the data, and this is the MLE.

Maximum likelihood estimation

- There is always a set of parameters that gives you the highest likelihood of observing the data, and this is the MLE.
- These can be calculated using:
 - Trial and error (not efficient!).
 - Compute the maximum of a function by hand (rarely doable in practice).
 - An iterative optimization algorithm: `?optim` in R.

By hand: compute MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

- $P(Y = k) = \binom{k}{N} p^k (1 - p)^{N-k} = L(p).$

By hand: compute MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

- $P(Y = k) = \binom{k}{N} p^k (1 - p)^{N-k} = L(p)$.
- $\log(L(p)) = \text{cte} + k \log(p) + (N - k) \log(1 - p)$.

By hand: compute MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

- $P(Y = k) = \binom{k}{N} p^k (1 - p)^{N-k} = L(p)$.
- $\log(L(p)) = \text{cte} + k \log(p) + (N - k) \log(1 - p)$.
- We are searching for the maximum of L , or equivalently that of $\log(L)$.

By hand: compute MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

- $P(Y = k) = \binom{k}{N} p^k (1 - p)^{N-k} = L(p)$.
- $\log(L(p)) = \text{cte} + k \log(p) + (N - k) \log(1 - p)$.
- We are searching for the maximum of L , or equivalently that of $\log(L)$.
- Compute derivate w.r.t. p : $\frac{d \log(L)}{dp} = \frac{k}{p} - \frac{(N - k)}{(1 - p)}$.

By hand: compute MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

- $P(Y = k) = \binom{N}{k} p^k (1 - p)^{N-k} = L(p)$.
- $\log(L(p)) = \text{cte} + k \log(p) + (N - k) \log(1 - p)$.
- We are searching for the maximum of L , or equivalently that of $\log(L)$.
- Compute derivate w.r.t. p : $\frac{d \log(L)}{dp} = \frac{k}{p} - \frac{(N - k)}{(1 - p)}$.
- Then solve $\frac{d \log(L)}{dp} = 0$; the MLE is $\hat{p} = \frac{k}{N} = \frac{3}{10} = 0.3$.

By hand: compute MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

- $P(Y = k) = \binom{k}{N} p^k (1 - p)^{N-k} = L(p)$.
- $\log(L(p)) = \text{cte} + k \log(p) + (N - k) \log(1 - p)$.
- We are searching for the maximum of L , or equivalently that of $\log(L)$.
- Compute derivate w.r.t. p : $\frac{d \log(L)}{dp} = \frac{k}{p} - \frac{(N - k)}{(1 - p)}$.
- Then solve $\frac{d \log(L)}{dp} = 0$; the MLE is $\hat{p} = \frac{k}{N} = \frac{3}{10} = 0.3$.
- Here, the MLE is the proportion of observed successes.

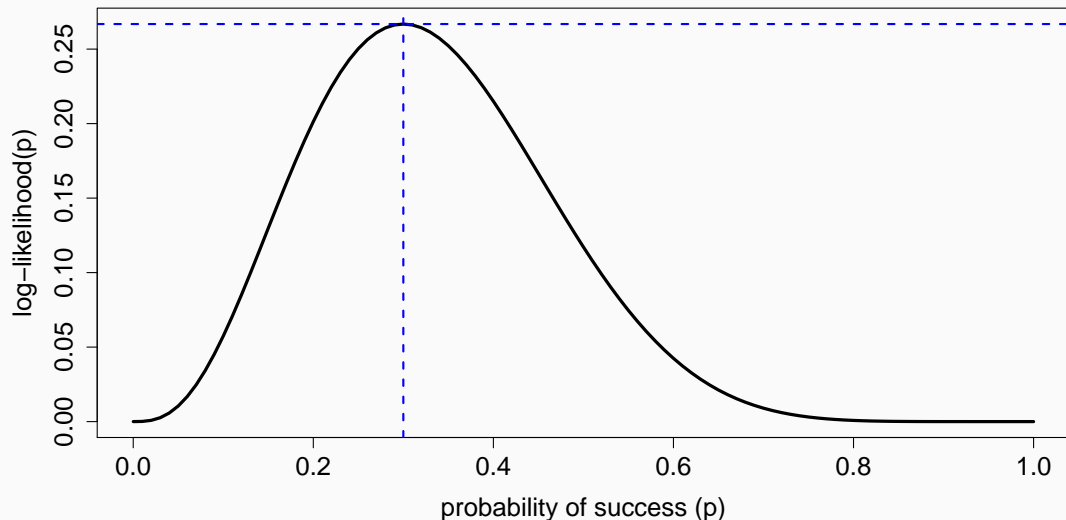
Using a computer: MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

```
lik.fun <- function(parameter) dbinom(x=3, size=10, prob=parameter)
# ?optimize
optimize(lik.fun, c(0,1), maximum=TRUE)
#> $maximum
#> [1] 0.3000157
#>
#> $objective
#> [1] 0.2668279
```

Use `optim` when the number of parameters is > 1 .

Using a computer: MLE of p from $Y \sim \text{Bin}(N = 10, p)$ with $k = 3$ successes

Binomial likelihood with 3 successes out of 10 attempts

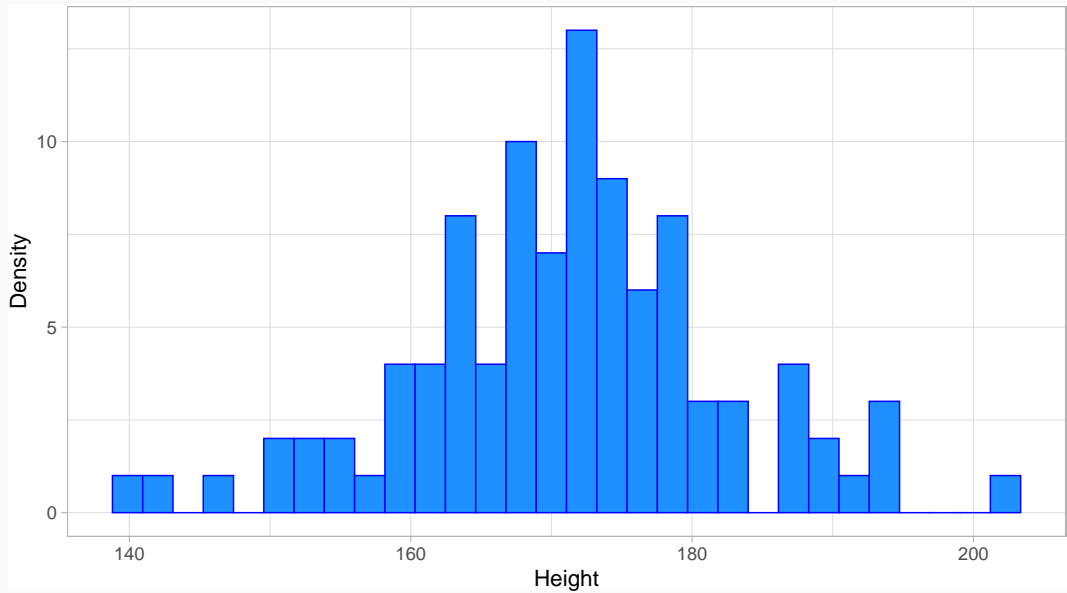


Your turn

MLE of the parameters of a Normal distribution

- Assume we have collected data on the height of 100 people:

```
# set seed for random numbers  
set.seed(2020)  
# simulate data from Normal distribution  
n <- 100  
height <- rnorm(n, mean=170, sd=10)
```



- We consider a Normal distribution for the model.
- Compute the MLE of the parameters of the Normal distribution.
- Hint: Use functions `optim()` and `dnorm()`

Solution

- Write a function for the likelihood of a Normal distribution with parameters mean μ and standard deviation σ :

```
negloglik <- function(theta, data) {  
  mu <- theta[1]  
  sigma <- theta[2]  
  x <- data  
  -sum(dnorm(x, mean = mu, sd = sigma, log = TRUE))  
}  
negloglik(theta = c(150,1), height)  
#> [1] 28530.45
```

- Minimiiiiize

```
fit <- optim(par = c(1,1), fn = negloglik, data = height)
```

```
fit
```

```
#> $par
```

```
#> [1] 171.05358 11.16656
```

```
#>
```

```
#> $value
```

```
#> [1] 382.9207
```

```
#>
```

```
#> $counts
```

```
#> function gradient
```

```
#>      135      NA
```

```
#>
```

```
#> $convergence
```

```
#> [1] 0
```

```
#>
```

