# **Ejercicio – Estimación básica de máxima verosimilitud**

## A)

Queremos estimar la prevalencia de un parásito en una población. Llamemos a este parámetro desconocido 𝑝.

* ¿Cuál es la probabilidad de que un individuo aleatorio de esta población no esté parasitado (expresada en función de 𝑝)?
* ¿Cuál es la probabilidad de que tres individuos al azar, muestreados de forma independiente, estén todos parasitados?
* Si examinas dos individuos al azar, ¿cuál es la probabilidad de que el primero esté parasitado y el segundo no?
* Si examina dos individuos al azar, ¿cuál es la probabilidad de que uno esté parasitado y el otro no?
* Si examina 10 individuos al azar, muestreados independientemente de la población, ¿cuál es la probabilidad de que 2 de ellos estén parasitados? (no tienes que dar una expresión exacta)

B)

En una muestra de n = 10 individuos se parasitan k = 4 individuos. La probabilidad de obtener estos datos en un orden particular (por ejemplo, obtener primero 4 individuos parasitados, luego 6 individuos no parasitados) es , y la probabilidad de obtener estos datos en cualquier orden es  (esta es la probabilidad binomial). La primera fracción, , es solo el número de secuencias diferentes que puede obtener 4 individuos parasitados en una muestra de 10 y, a menudo, se reduce a .

* Calcula la probabilidad de obtener estos datos dado que 𝑝 = 0, 0.1, …, 1 (en R puedes usar la función ‘dbinom(k,n,p)’). Luego haz un gráfico de esta probabilidad como función de 𝑝.
* Prueba diferentes valores de tamaño de muestra (𝑛) y número de individuos parasitados (𝑘) y observa cómo cambia el gráfico. ¿Cómo puede usar este gráfico para encontrar la “mejor suposición” de la prevalencia del parásito en toda la población con base en estos datos?
* Intenta aumentar tanto el tamaño de la muestra (𝑛) como el número de individuos parasitados (𝑘) manteniendo las mismas proporciones (ej., k/n = 20/8, 30/12, 40/16,…). ¿Cómo cambia la forma de la curva cuando aumenta el tamaño de la muestra? como interpretas esto?
* Calcula también la probabilidad de los datos dados los diferentes valores de 𝑝 cuando los individuos parasitados aparecen en un orden particular (p. ej., primero obtienes 4 parasitados, luego 6 no parasitados) y grafica esto. ¿En qué se diferencian las dos curvas? ¿Esto tiene sentido? ¿Obtenemos alguna información sobre el parámetro 𝑝 sabiendo el orden en el que tomamos muestras de individuos parasitados y no parasitados?

El último gráfico que hiciste se llama función de verosimilitud de 𝑝 dados los datos (𝑘 y 𝑛) y se escribe . Tenga en cuenta que esta es la misma expresión que la función de probabilidad binomial, excepto que vemos la expresión como una función de 𝑝 en lugar de una función de 𝑘 y nos hemos saltado el coeficiente binomial porque es solo una constante que solo afecta la elevación de la curva (no la forma o la ubicación del pico). El valor de los parámetros que maximizan la función de verosimilitud se denomina estimación de máxima verosimilitud (a menudo abreviado como MLE). Esta es su "mejor suposición" para el valor del parámetro. En este caso podemos escribir 𝑝̂ 𝑀𝐿𝐸 = 0.4 (el 'sombrero' sobre la 𝑝 indica que se trata de una estimación del parámetro).

* Usually, the logarithm of the likelihood function, the log-likelihood, is used to find the MLE. The log-likelihood function is in this case ℓ(𝑝; 𝑘, 𝑛) = 𝑘 ln(𝑝) + (𝑛 − 𝑘)ln (1 − 𝑝). The log-likelihood function and the likelihood function always have the peak for the same value of the parameter(s). You can confirm this by plotting the log-likelihood function in the spread sheet as well. C) In the simple example above you can find the MLE analytically by finding the value of the parameter where the likelihood function is flat (i.e., at the peak), by setting the first derivative to zero, 𝜕ℓ 𝜕𝑝 = 0, and solving this for 𝑝. You would then get 𝑝̂ 𝑀𝐿𝐸 = 𝑘/𝑛. However, in more complex models this can often not be done analytically so it has to be done by “trail-and-error” numerically (which is often a lot easier anyway). There are several such functions for such numerical optimization in R (e.g., ‘optim’ and ‘optimize’). It is also possible to do numerical optimization in Excel: n k p Binomial probability Likelihood Log likelihood 10 4 0 0 0 0.1 0.011160261 5.31441E-05 -9.8425 0.2 0.088080384 0.00041943 -7.77661 0.3 0.200120949 0.000952957 -6.95594 0.4 0.250822656 0.001194394 -6.73012 0.5 0.205078125 0.000976563 -6.93147 0.6 0.111476736 0.000530842 -7.54105 0.7 0.036756909 0.000175033 -8.65054 0.8 0.005505024 2.62144E-05 -10.5492 0.9 0.000137781 6.561E-07 -14.237 1 0 0 0 0014 Likelihood 0 0.05 0.1 0.15 0.2 0.25 0.3 0 0.2 0.4 0.6 0.8 1 Binomial probability Binomial probability 0 Log likelihood - Write an arbitrary value for 𝑝 in one cell, and the likelihood function or log-likelihood function based on this 𝑝 in another cell. - In Excel 2007, go to “Problem solver” under the ‘Data’ tab. In older versions of Excel go to ’Tools > Solver’ (if you haven’t used Solver before, you probably need to go to ’Tools > Add-Ins’ and select “Solver Add-in”). - Set up solver to maximize the likelihood by changing the value in the 𝑝-cell. - Try various numbers of 𝑛 and 𝑘 To do this exercise in R, you can use the following code: N = 10 K = 4 # The likelihood function L = function(p,k,n) p^k\*(1-p)^(n-k) # The log-likelihood function l = function(p,k,n) k\*log(p) + (n-k)\*log(1-p) # Plotting the binomial probability of k given p p = seq(0,1,0.001) plot(p, dbinom(k,n,p), type="l") # Plotting the Likelihood function in a new window windows() plot(p, L(p,K,N), type="l") # Plotting the Likelihood function in a new window windows() plot(p, l(p,K,N), type="l") # The optimization functions in R finds the minimum, not the maximum. We # therefor must create new functions that return the negavive likelihood and # log-likelihood, and then minimize these: # Minus likelihood: mL = function(p,k,n) -p^k\*(1-p)^(n-k) # minus log-likelihood: ml = function(p,k,n) -(k\*log(p) + (n-k)\*log(1-p)) # Using 'optimize' (NB! if you have more than one parameter, you should use # 'optim' optimize(mL, interval = c(0,1), k=K, n=N) optimize(ml, interval = c(0,1), k=K, n=N) K/N