Profesor: Héctor Bahamonde, PhD.

e:hector.bahamonde@uoh.cl

w:www.hectorbahamonde.com

Curso: MLE.

TA: Gonzalo Barría.

I. Likelihood e Inferencia

Recuerda que Equation 1 es el "axiom likelihood". Lo que dice que likelihood es "proporcional" a la probabilidad (King 1998, p. 59). Y donde también la constante $k(y)$ asegura que el likelihood es relativo al modelo, y no un absoluto como ocurre en el paradigma de la probabilidad.

$$\begin{aligned} L(\tilde{\theta}|y) &= k(y)Pr(y|\tilde{\theta}) \\ &\propto Pr(y|\tilde{\theta}) \end{aligned} \tag{1}$$

Ahora, tratemos de estimar nuestro parámetro β pero usando MLE. Es decir, en vez de estimar $\beta = (x^\top x)^{-1}x^\top y$, lo estimaremos usando la Equation 1 de la siguiente manera:

$$\begin{aligned} L(\tilde{\beta}|y) &= k(y)Pr(y|\tilde{\beta}) \\ &= k(y)\prod_{i=1}^{n} f(y_i|\tilde{\beta}) \\ &\propto \prod_{i=1}^{n} f(y_i|\tilde{\beta}) \\ &= \prod_{i=1}^{n}(2\pi)^{-\frac{1}{2}}\exp\left[\frac{-(y_i-\beta)^2}{2}\right] \end{aligned} \tag{2}$$

Importantemente, Equation 2 nos entrega el likelihood relativo de que **el modelo (capturado por β) produzca los datos** y_i. Sobre esto, hay que rescatar los siguientes puntos:

1. MLE es inverso a la probabilidad: en MLE el modelo es construido, y los datos son "dados".
2. Para hacer obtener el likelihood (que es simplemente un número), trabajaremos con el conocido "log-likelihood function", que es básicamente sacar el log de Equation 2 (tal como lo demuestra

Equation 3).

3. Usamos *product operators* ($\prod$) porque necesitamos multiplicar el likelihood de cada i particular (cada fila en la columna y). Esto es una consecuencia de que por ejemplo i_1 es independiente a i_2 y asi para todos los i_n.

Veamos ahora el "log-likelihood function":

$$\begin{aligned} L(\tilde{\beta}|y) &= k(y)Pr(y|\tilde{\beta}) \\ ln\ L(\tilde{\beta}|y) &= ln\left\{k(y)Pr(y|\tilde{\beta})\right\} \\ ln\ L(\tilde{\beta}|y) &= -\frac{1}{2}\sum_{i=1}^{n}(y_i-\tilde{\beta})^2 \end{aligned} \tag{3}$$

Debido a que el logaritmo de un producto ($k(y)Pr(y|\tilde{\beta})$) es la suma de los logaritmos, y además de que podemos usar el *Fisher-Neyman Factorization Lemma*, la tercera línea es la forma simplificada del "log-likelihood function". Nota que $ln\ L(\tilde{\beta}|y)$ depende de los errores cuadrados, es decir, la distancia entre lo predicho y lo observado, i.e. $(y_i-\tilde{\beta})^2 = \epsilon_i^2$.

MLE y OLS Veámos las diferencias, si es que las hay, entre los métodos de estimación OLS y MLE.

```
# cocinemos los datos
n <- 1000
x1 <- rnorm(n, mean = 150, sd = 3)
x2 <- rnorm(n, mean = 100, sd = 2)
e  <- rnorm(n, 0)
y  <- 5 + 2*x1 + 3*x2 + e
# OLS:
ols <- lm(y ~ x1 + x2)
# GLS
mle <- glm(y ~ x1 + x2, family=gaussian)
# Tabla
```

```
p_load(texreg)
screenreg(l = list(ols, mle), custom.model.names=c("OLS", "MLE"))

##
## ========================================
##                 OLS          MLE
## ----------------------------------------
## (Intercept)        2.91          2.91
##                   (2.17)        (2.17)
## x1                 2.01 ***      2.01 ***
##                   (0.01)        (0.01)
## x2                 3.01 ***      3.01 ***
##                   (0.02)        (0.02)
## ----------------------------------------
## R^2                0.99
## Adj. R^2           0.99
## Num. obs.       1000          1000
## AIC                           2861.06
## BIC                           2880.69
## Log Likelihood               -1426.53
## Deviance                      1015.30
## ========================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Como te darás cuenta, ambos métodos dan exactamente los mismos resultados. Es más, la última línea de la Equation 3 muestra que el log-likelihood $ln\ L(\tilde{\beta}|y)$ es completamente una función de y. Hasta el momento, nada ha cambiado desde OLS. Bajo ese paradigma, nos enfocábamos en estimar β, que no es nada más que el valor esperado de una variable mantenida en su *promedio*. En MLE, es bastante parecido. De hecho, King (1998, p. 68) explica que "the maximum likelihood estimator for β [...] also has a familiar analytical solution: $\beta = (x^\top x)^{-1}x^\top y$". En vez de β, en MLE nos concentramos en π que también es el valor esperado. Sin embargo, cambiamos el modo de leer lo que significa ese valor esperado. Como dice King (1998, pp. 66–67), los estimadores MLE

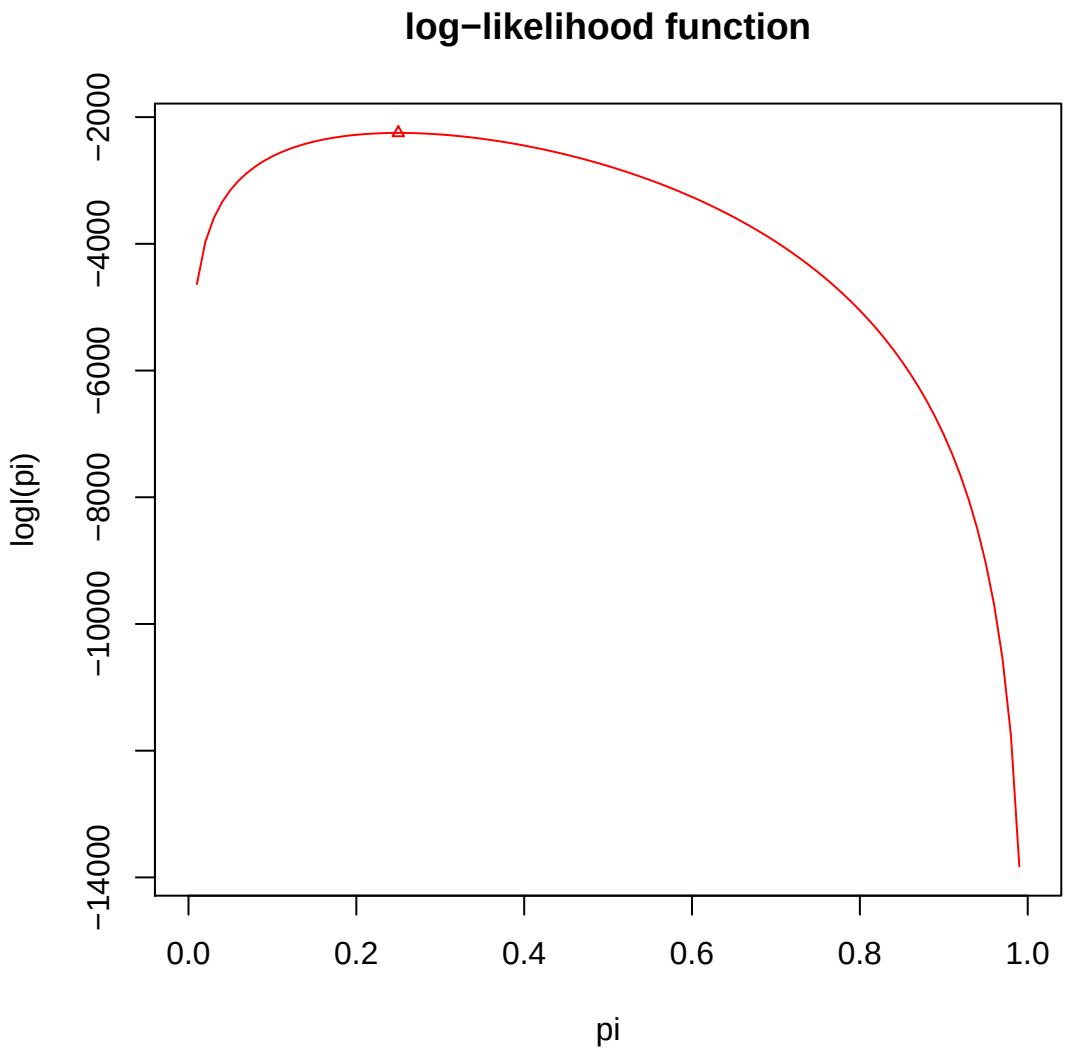
tienen "the highest relative likelihood of having generated the data we observed".

Por qué usar otro método entonces? En este caso, nosotros sabemos (porque así lo hemos construido), que `e` tiene promedio 0 (`e <- rnorm(n, 0)`). Ya sabemos que este es un "population parameter" y que nunca llegaremos a saber si es el promedio es cero ($E(e) = 0$ es un supuesto). Entonces, *por qué usar otro método entonces?* Aunque no sabemos si los supuestos se cumplen en la regresión lineal, *sí sabemos* que estos supuestos **no se cumplen** en modelos cuyas variables dependientes binarias o categoricas.

```
# cocinemos los datos
n <- 1000 # sample size es 1000
ym = 3 # promedio de y es 3.
# Funcion para encontrar el LL
pi=c(1:100)/100
logl=function(pi) n*(ym*log(1-pi)+log(pi))
score=function(pi) n*(1/pi-ym/(1-pi))
logl1=n*(ym*log(1-pi)+log(pi))
score1=n*(1/pi-ym/(1-pi))
```

Cómo se ve el log-likelihood? Grafiquemos.

```
# Grafico
plot(logl, col="red",xlab="pi",ylab="logl(pi)")
points(pi[score1==0],logl1[pi==pi[score1==0]],pch=2,col="red",cex=0.6)
title("log-likelihood function")
```



Como verás, cuando $E(y) = \pi = 3$, el **log-likelihood**, es decir, **el valor que maximiza la posibilidad de que la distribucion de y tenga un promedio 3** es $ln\ L(\hat{\theta}|y) = -2249.3$ (fíjate que reemplazamos β por $\hat{\theta}$ para volver a nuestra notación MLE). Veámos:

```
round(max(logl1),1)

## [1] -2249.3
```

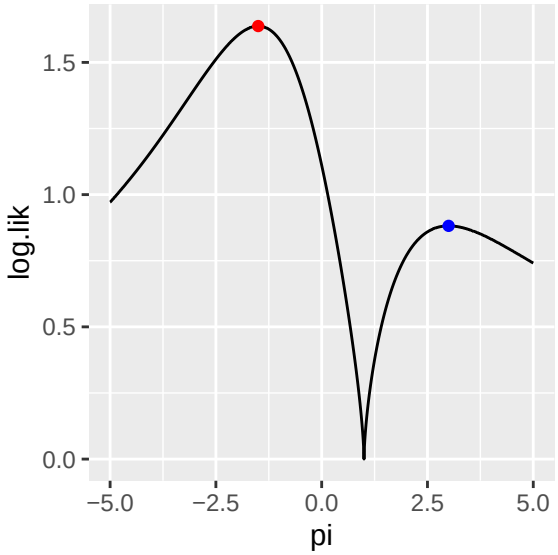
Al margen de los códigos de R, vemos claramente que -2249.3 es el log-likelihood, i.e. el valor que hace que y tenga promedio 3. *Cómo se calcula el log-likelihood?*

II. Métodos Analíticos

1. Calcular la derivada de la función log-likelihood con respecto al vector de parámetros $\tilde{\theta}$.
2. Setear la derivada a 0, y substituir cada elemento de $\tilde{\theta}$ con $\hat{\theta}$.
3. Si es posible, resolver por $\hat{\theta}$.
4. Tomar la segunda derivada de la función log-likelihood. Si es negativa, $\tilde{\theta}$ es el maximum likelihood estimator.

Este procedimiento asegura que caigamos en el global maxima y no en el local maxima.

```
p_load(ggplot2,ggpmisc)
pi <- seq(-5,5,length=10001)
log.lik <- (10*((pi-1)^2)^(1/3))/(pi^2 + 9)
df <- data.frame(pi = pi, log.lik = log.lik)
ggplot(data = df, aes(x = pi, y = log.lik)) + geom_line() + stat_peaks(col = c("red","blue"))
```



III. Métodos Numéricos

Muchas veces resolver por $\hat{\theta}$ es algebraicamente imposible (King 1998, p. 72). Es por esto que este proceso empieza por seleccionar "valores de partida" (*starting values*), y a través de la especificación de los distintos valores de que puede tomar el parametro $\tilde{\theta}$, vamos viendo si llegamos al global maxima. Esto es justamente lo que se hizo con el "grid search" arriba (uno de los dos métodos numéricos), y se hace para toda la superficie de la distribución. El segundo método numérico es el método de "gradiente". Este método usa la pendiente (slope) para ver cuando la pendiente es cero. En general, nadie usa métodos así. El computador se encarga de maximizar, pero es importante saber qué es lo que está haciendo el programa.

References

King, Gary (1998). *Unifying Political Methodology: The Likelihood Theory of Statistical Inference*. Ann Arbor, MI: University of Michigan Press, pp. 1–274.