



Distribución de Poisson: interpretación de los resultados (III)

Tras analizar los datos que provienen de contajes con la distribución de Poisson y de datos que provenían de un análisis de lechones nacidos muertos, en esta entrega llega la hora de descifrar los resultados.

Comenzamos en el número de enero de 2023 (*Suis* 194) una serie sobre el análisis de datos que provienen de contajes con la distribución de Poisson y en el número de febrero de 2023 (*Suis* 195) analizamos unos datos que provenían de un análisis de lechones nacidos muertos que se encuentra en nuestro blog (<https://www.testsandtrials.com/blog/>) con el nombre de “data_poisson.txt”.

Para seguir con el ejercicio debemos cargar los datos en RCommander y repetir lo que hicimos en el número anterior yendo a Estadísticos/Ajustes de modelos/Modelo lineal generalizado eligiendo la variable nac_muer como variable respuesta y como predictoras ya vimos que solamente usaremos la variable genética y la variable peso (ver *Figura 1*).

Y la salida que obtenemos es esta ocasión es la que se muestra en la columna de la derecha:

```
Call:
glm(formula = nac_muer ~ genética+peso, family = poisson(log),
     data = data)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4779 -1.2706 -0.7908  0.5627  3.6575
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.12167   0.08284  13.540 < 2e-16 *
genética[T.B] -1.33584   0.09398 -14.214 < 2e-16 *
peso[T.muy gorda] -0.40664   0.11116  -3.658 0.000254 *
peso[T.normal]  -0.94845   0.1149  -8.249 < 2e-16 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 1052.95 on 510 degrees of freedom
Residual deviance: 805.14 on 507 degrees of freedom
AIC: 1326.8
```

Alberto Morillo Alujas¹,
Daniel Villalba Mata² y
Emilio López Cano³

¹Tests and Trials SLU
²Universidad de Lleida
³Universidad Rey Juan Carlos (Madrid)

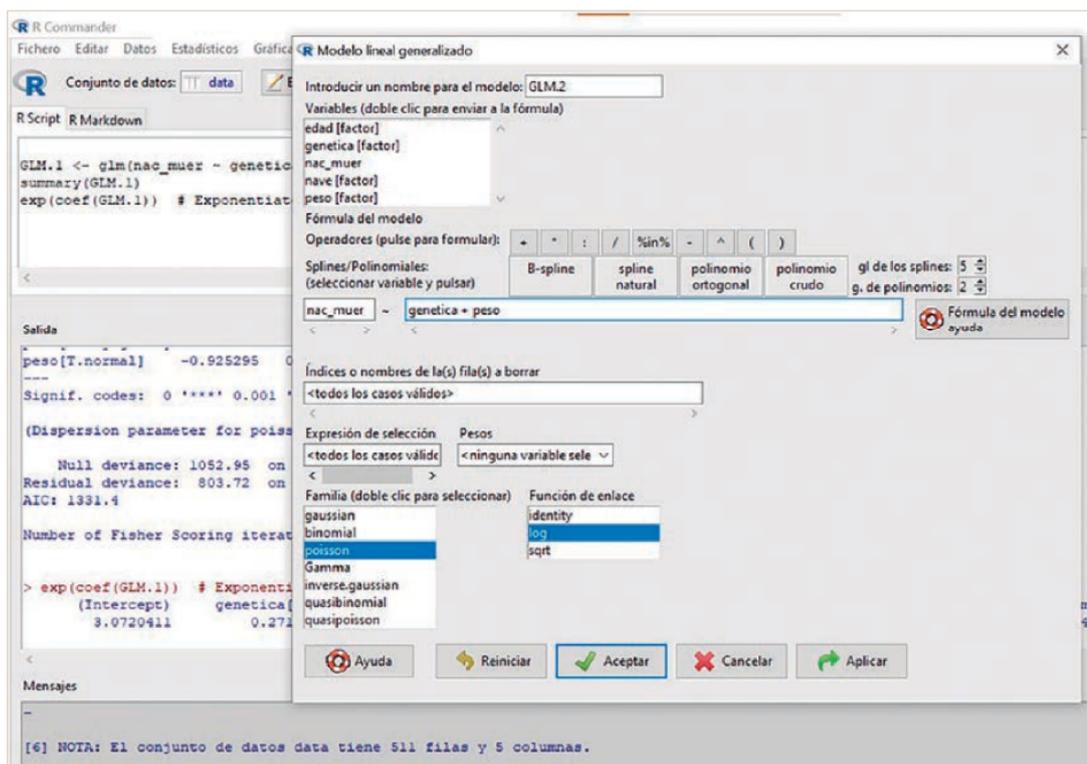


Figura 1: GLM sin las variables edad y nave.

En la salida que hemos marcado en cursiva, ver página anterior, tenemos los coeficientes de la ecuación del modelo. Así a diferencia de la regresión lineal donde la variable respuesta viene definida por una ecuación del tipo:

$$Y = \alpha + \beta_1 x_1 + \dots + \beta_n x_n + \epsilon \quad (1)$$

en el modelo de regresión de Poisson la variable respuesta viene definida por:

$$\log(Y) = \alpha + \beta_1 x_1 + \dots + \beta_n x_n + \epsilon \quad (2)$$

ya que, como el lector recordará, al construir el modelo hemos elegido la familia Poisson y la función de enlace era el logaritmo (log). Por lo que los valores de los coeficientes del modelo son logarítmicos siendo la ecuación 2 equivalente a:

$$Y = e^{(\alpha + \beta_1 x_1 + \dots + \beta_n x_n + \epsilon)} \quad (3)$$

o lo que es lo mismo y simplificando a dos variables predictoras (x_1 y x_2 , genética y Peso en nuestro ejemplo):

$$Y = e^{\alpha} + e^{\beta_1 x_1} + e^{\beta_2 x_2} \quad (4)$$

Pero para comprender las diferencias en nuestro modelo, vamos a convertir estos logaritmos. Si bien la salida de RCommander ya nos informa de ello al final:

```
> exp(coef(GLM.2)) # Exponentiated coefficients
(Intercept) genetica[T.B] peso[T.muy gorda] peso[T.normal]
3.0699624    0.2629371      0.6658841      0.3873398
```

donde los coeficientes han sido exponentiados, es mucho más fácil y completo tener todos los coeficientes, sus errores estándar y sus intervalos de

confianza en una tabla. Para ello tendremos que usar la función emmeans escribiendo unas líneas de código muy simples en la consola, ya que RCommander no habilita esta función. Así, escribiremos:

```
salida <- emmeans(GLM.2, pairwise~genetica,
type="response", weights = "proportional")
summary(salida, infer = TRUE)
```

y obtendremos la siguiente salida:

```
$emmeans
genetica rate SE df asymp.LCL asymp.UCL null z.ratio
p.value
A 1.881 0.1192 Inf 1.661 2.13 1 9.965 <.0001
B 0.494 0.0361 Inf 0.429 0.57 1 -9.657 <.0001
```

Results are averaged over the levels of: peso
Confidence level used: 0.95
Intervals are back-transformed from the log scale
Tests are performed on the log scale

```
$contrasts
contrast ratio SE df asymp.LCL asymp.UCL null
z.ratio p.value
A/B 3.8 0.357 Inf 3.16 4.57 1 14.214
<.0001
```

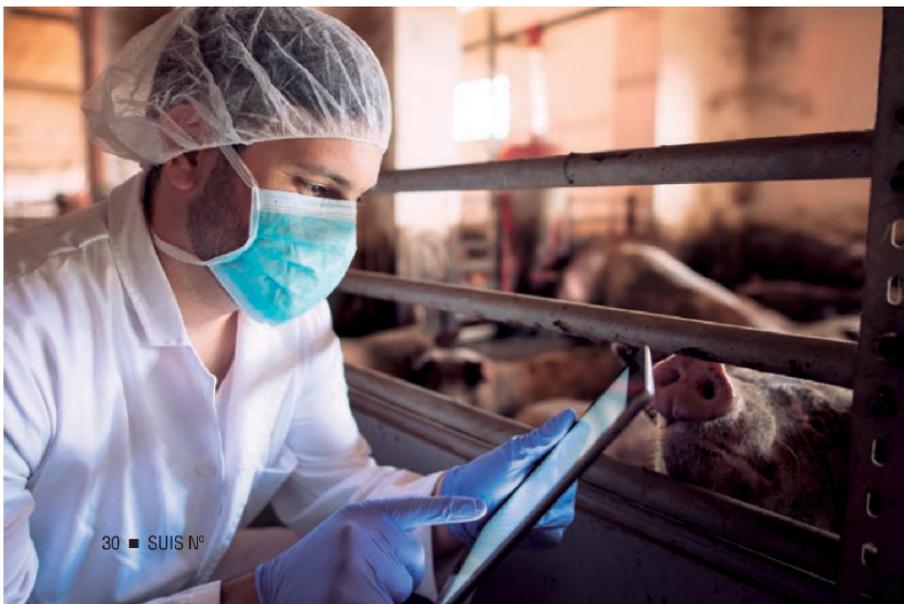
Results are averaged over the levels of: peso
Confidence level used: 0.95
Intervals are back-transformed from the log scale
Tests are performed on the log scale

Ahora es fácil llegar a la interpretación de nuestros resultados. En concreto, en la primera tabla (\$emmeans) encontramos que la genética A tiene de media 1,881 lechones nacidos muertos con un intervalo de confianza entre 1,661 y 2,13, mientras que la genética B tiene 0,494 con un intervalo de confianza entre 0,429 y 0,57 lechones nacidos muertos, siendo la diferencia muy significativa como se puede comprobar en la probabilidad del test en el apartado "contrasts". Este apartado también nos proporciona la tasa o ratio entre las dos genéticas, que nos dice que la genética A tiene 3,8 veces más lechones nacidos muertos que la B, con un intervalo de confianza entre 3,16 y 4,57 veces. Esta es una gran utilidad de los modelos de Poisson. No solo podemos ver la diferencia, sino también la proporción o ratio entre los valores.

Estos resultados están equilibrados por los niveles de peso, sin embargo, también podemos obtener los valores de los lechones nacidos muertos en las diferentes genéticas y en los diferentes pesos. Si escribimos en la consola las instrucciones:

```
salida2 <- emmeans(GLM.2, pairwise~genetica+peso,
type = "response", weights = "proportional")
summary(salida2, infer = TRUE)
```

obtenemos la siguiente salida que nos da los lechones muertos por genética y peso:



Aleksandar Malivuk/Shutterstock.com

```
Semmeans
genetica peso rate SE df asymp.LCL asymp.UCL
null z.ratio p.value
A gorda 3.070 0.2543 Inf 2.610 3.611 1
13.540 <.0001
B gorda 0.807 0.0670 Inf 0.686 0.950 1
-2.578 0.0099
A muy gorda 2.044 0.1962 Inf 1.694 2.467 1
7.450 <.0001
B muy gorda 0.538 0.0553 Inf 0.439 0.658 1
-6.036 <.0001
A normal 1.189 0.1166 Inf 0.981 1.441 1 1.766
0.0773
B normal 0.313 0.0341 Inf 0.252 0.387 1
-10.656 <.0001
```

Confidence level used: 0.95
 Intervals are back-transformed from the log scale
 Tests are performed on the log scale

```
$contrasts
contrast ratio SE df asymp.LCL asymp.UCL
null z.ratio p.value
A gorda/B gorda 3.803 0.3574 Inf 2.910 4.971 1
14.214 <.0001
A gorda/A muy gorda 1.502 0.1669 Inf 1.094 2.061
1 3.658 0.0035
A gorda/B muy gorda 5.711 0.8571 Inf 3.724 8.759
1 11.612 <.0001
A gorda/A normal 2.582 0.2968 Inf 1.860 3.583
1 8.249 <.0001
A gorda/B normal 9.819 1.5307 Inf 6.297 15.311
1 14.653 <.0001
B gorda/A muy gorda 0.395 0.0556 Inf 0.264 0.590
1 -6.594 <.0001
B gorda/B muy gorda 1.502 0.1669 Inf 1.094 2.061
1 3.658 0.0035
B gorda/A normal 0.679 0.0955 Inf 0.455 1.014
1 -2.753 0.0653
B gorda/B normal 2.582 0.2968 Inf 1.860 3.583
1 8.249 <.0001
A muy gorda/B muy gorda 3.803 0.3574 Inf 2.910
4.971 1 14.214 <.0001
A muy gorda/A normal 1.719 0.2179 Inf 1.198 2.467
1 4.275 0.0003
A muy gorda/B normal 6.538 1.0505 Inf 4.136 10.335
1 11.686 <.0001
B muy gorda/A normal 0.452 0.0700 Inf 0.291 0.703
1 -5.128 <.0001
B muy gorda/B normal 1.719 0.2179 Inf 1.198 2.467
1 4.275 0.0003
A normal/B normal 3.803 0.3574 Inf 2.910 4.971
1 14.214 <.0001
```

Confidence level used: 0.95
 Conf-level adjustment: tukey method for comparing a family of 6 estimates
 Intervals are back-transformed from the log scale
 P value adjustment: tukey method for comparing a family of 6 estimates
 Tests are performed on the log scale

En esta segunda salida, en el apartado \$contrasts nos proporciona la comparación (con su tasa, intervalo de confianza y significación) de todas las combinaciones posibles entre los dos factores que estamos estudiando.

Podemos obtener también un gráfico que nos facilitará la comprensión, cargando la librería ggplot2 que ya hemos usado algunas veces y ejecutando una instrucción muy simple:

```
library(ggplot2)
emmip(salida2, ~ genetica | peso, CIs = TRUE) + facet_
wrap(~ peso) + labs(x = "Genética," y = "Lechones nacidos
muertos")
```

y obtendremos el siguiente gráfico:

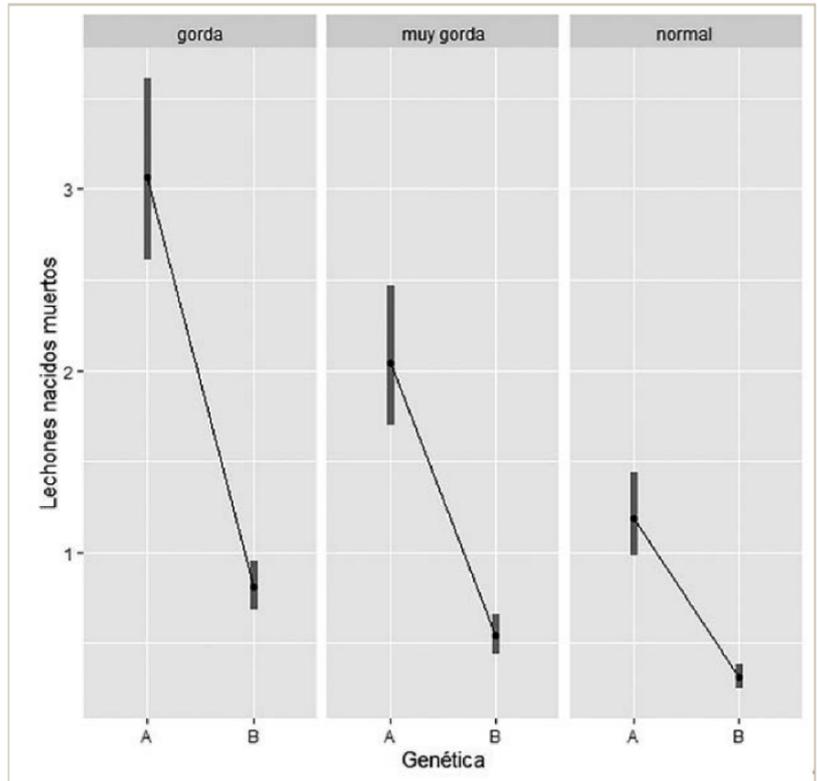


Figura 2: Lechones nacidos muertos por genética y peso de la cerda al parto con el modelo Poisson.

Cuando hemos escrito por primera vez en la consola la instrucción:

```
salida <- emmeans(GLM.2, pairwise~genetica,
type="response", weights = "proportional")
summary(salida, infer = TRUE)
```

hemos usado el argumento weights = "proportional", ya que al iniciar este estudio observacional hemos creado unas tablas y hemos visto que no había el mismo número de cerdas ni en cada genética, ni nave, ni distribuidas por peso. Si no lo hubiéramos usado, la función emmeans por defecto balancea las medias como si en todos los grupos hubiera el mismo número de observaciones. Si el lector ejecuta la instrucción sin ese argumento weights observará las ligeras diferencias.

En la función summary, hemos usado el argumento infer para obtener tanto los intervalos de confianza como las significaciones. El lector puede ver qué obtiene si no se indica este argumento.