Ser o no ser: variables aleatorias discretas binarias, resultado e interpretación (II)

En el último capítulo (Suis nº 198, junio 2023) estudiamos un tipo especial de variables respuesta, las variables aleatorias discretas binarias, que suelen ser atributos y cuyo valor se reduce a dos posibilidades.

Alberto Morillo Alujas[1], Daniel Villalba Mata[2] y Emilio López Cano[3]

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

En la anterior entrega se dijo también que el resumen de las variables binarias se expresa en forma de proporciones, ya que resume de una forma rápida cuántos objetos experimentales hay en una categoría, por ejemplo, muertos, machos, sanos, etc., pero también, y esto es primordial, cuántos hay en la otra categoría, por ejemplo, vivos, hembras, enfermos, etc.

De la misma manera, vimos también muy someramente a qué llamamos probabilidad, siendo "p" la probabilidad de que aparezca un suceso de interés o éxito y "q" la probabilidad de que no aparezca y que la probabilidad máxima es 1, debido a que la suma de "p+q = 1".

Otra cuestión que se abordó fue que la media de una variable aleatoria discreta binomial, que cuenta el número de éxitos en muestras de tamaño n, es el producto de n por p "np" y que su varianza es particular, ya que depende de la media. La varianza es baja cuando "p" es o muy alta (de 0,8 en adelante) o es muy baja (de 0,2 hacia abajo) y es máxima cuando "p" (y por lo tanto también "q") es igual a 0,5.

Y comenzamos a usar el archivo *germination.txt* del libro "The R Book" de Michael J. Crawley (2013) que puede descargarse en *www.testsandtrials.com/blog* con el nombre *orobanche.xlsx* debidamente modificado.

Una vez activado R, cargaremos nuestro ayudante *RCommander* mediante la instrucción *library*(Rcmdr) e importaremos el archivo *orobanche.xlsx* yendo a *Datos/Importar datos/desde un archivo Excel.* y le daremos el nombre "orobanche". Elegiremos nuestro archivo y veremos qué datos contiene yendo a *Estadísticos/Resúmenes/Conjunto de datos activos*.

```
> summary(orobanche)
Celo              tot_nul          granja    tratamiento
 Min.   : 0.00    Min.   : 4.00    a73:10    a:10
 1st Qu.: 8.00    1st Qu.: 16.00   a75:11    b:11
 Median: 17.00    Median: 39.00
 Mean   : 20.19   Mean   : 39.57
 3rd Qu.: 26.00   3rd Qu.: 51.00
 Max.   : 55.00   Max.   : 81.00
```

Los datos podían ser el resultado de un tratamiento con dos protocolos (a y b) en 21 lotes de cerdas nulíparas que están en dos granjas diferentes (a73 y a75) y hemos estudiado su salida a celo. La variable "celo" nos indica el número de cerdas nulíparas en celo y la variable "tot_nul" el número inicial de nulíparas tratadas. Nuestra hipótesis es que el tratamiento no tiene un resultado diferente en las granjas que hemos usado, es decir, no hay interacción entre tratamiento y granja.

Para estudiar variables binarias en R dijimos que debemos indicarle en la variable respuesta el número de éxitos y el número de fracasos, es decir, "p" y "q". Esto lo hacemos construyendo una variable respuesta con dos vectores. Usaremos una función llamada *cbind*, que nos unirá los éxitos y fracasos en un único objeto. Si el número de nulíparas en celo es la variable "celo", el número de nulíparas que no salieron en celo es igual al total de tratadas menos las que salieron en celo:

```
tot_nul– celo
```

y creamos una nueva variable que es nuestra variable de estudio y que denominaremos sencillamente "y". Lo hicimos escribiendo la siguiente instrucción en la consola y ejecutándola:

```
y <- cbind(celo, tot_nul- celo)
```

En nuestra hipótesis usamos un modelo lineal generalizado y dado que el modelo tenía sobredispersión[1] usamos la distribución quasibinomial que añade un parámetro adicional para describir la varianza adicional de los datos que no pueden explicarse mediante la binomial.

Para seguir con nuestro ejemplo iremos a *Estadísticos/Ajuste de modelos/Modelo lineal generalizado*, en el nombre del modelo, escribiremos "model1",

[1]Sobredispersión. Característica que nos indica que la variación es superior a la esperada y que podemos calcular dividiendo la desviación residual (*Residual deviance*) con los grados de libertad (*degrees of freedom*). Este resultado es casi 2 (1,957588) cuando debería no ser mayor que 1.

y especificaremos que nuestra variable respuesta es "y" en el constructor del modelo y usaremos las variables "granja" y "tratamiento" como predictoras, y elegiremos la familia quasibinomial como puede verse en la *figura 1*. Y obtendremos la salida:

```
Coefficients:
Estimate                 Std. Error  t value  Pr(>|t|)
(Intercept)              -0.4122     0.2513   -1.640   0.1193
granja[T.a75]            -0.1459     0.3045   -0.479   0.6379
tratamiento[T.b]          0.5401     0.3409    1.584   0.1315
granja[T.a75]:
tratamiento[T.b]          0.7781     0.4181    1.861   0.0801 .
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be
1.861832)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 33.278  on 17  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

Podemos observar que la significación de la interacción entre "granja" y "tratamiento" ha desaparecido (p = 0,0801), aunque todavía mantenemos sobredispersión que trataremos de corregir, creando un tercer modelo ya sin interacción.
Crearemos el nuevo modelo yendo de nuevo a *Estadísticos/Ajuste de modelos/Modelo lineal generalizado* y ahora en vez de analizar el modelo completo (con el asterisco entre las variables granja y tratamiento), elegiremos el modelo sin interacción, como en la *figura 2*, con el signo (+) entre las variables granja y tratamiento.
Obtendremos la siguiente salida:

```
> summary(model2)
Coefficients:
Estimate                 Std. Error  t value  Pr(>|t|)
(Intercept)              -0.7005     0.2199   -3.186   0.00512**
granja[T.a75]             0.2705     0.2257    1.198   0.24635
tratamiento[T.b]          1.0647     0.2104    5.061
0.0000814 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be
2.128368)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 39.686  on 18  degrees of freedom
AIC: NA
```

Por el valor de "p" en la variable "granja" ya podemos deducir que el tratamiento no tuvo ninguna diferencia entre las granjas (p = 0,246), pero vemos que el tratamiento "b" es diferente del tratamiento "a" (p = 0,0000814) con lo que el tratamiento sí afecta al número de nulíparas que salen en celo. Por otra parte, vemos que la sobredispersión está controlada (si dividimos la *residual deviance*, 39.686, entre los *degrees of freedom*, 18 el valor es cercano a 2).
Podríamos dejar así el modelo, pero técnicamente debemos eliminar la variable "granja" del mismo y tendremos el modelo 3 (*model3*) que construiremos eliminando la variable "granja" de nuestro ayudante. El resultado ahora será:

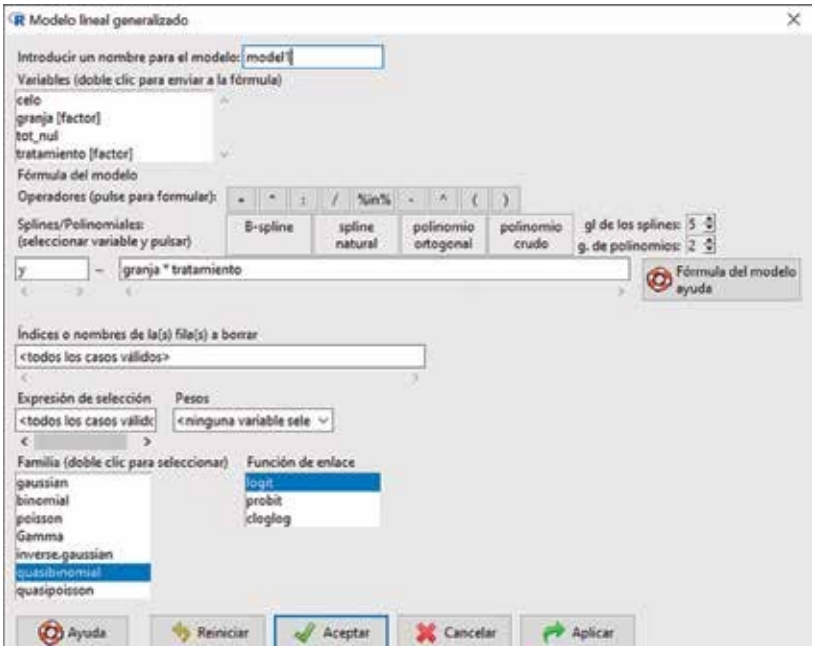

Figura 1. Construcción del modelo quasibinomial.

```
Coefficients:
Estimate                 Std. Error  t value  Pr(>|t|)
(Intercept)              -0.5122     0.1531   -3.345   0.0034 **
tratamiento[T.b]          1.0574     0.2118    4.992   0.0000809***
--
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be
2.169821)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 42.751  on 19  degrees of freedom
AIC: NA
```

Pero ¿cómo interpretamos esta salida? Hay que recordar que estamos tratando un modelo con los errores distribuidos según la ley binomial y que estos están en *logits*. Y recordar que en este caso:

$$p = \frac{1}{1+1/ex}$$

Para retornar los valores de *logits* a una escala comprensible debemos transformar cada "x" en una proporción "p" mediante la ecuación anterior. R lo hace por nosotros, pero vamos a entender cómo se calcula. Lo primero transformaremos el valor del "intercept *logit*" que es la proporción de nulíparas en celo con el tratamiento "a". Este valor es igual a (-0,5122). Si escribimos en la consola de R:

```
1/(1+1/(exp(-0.5122))),
```

nos devolverá el valor 0,3746779, igual a 37,47 %, que es el porcentaje de nulíparas en celo con el tratamiento "a". El valor para el tratamiento "b", tiene parecido tratamiento, pero hay que recordar que tenemos que sumar el valor de "b" con el de "a". Así escribiremos:

```
1/(1+1/(exp(-0.5122+ 1.0574))),
```

que nos dará 0,6330212, que en porcentaje es 63,30 %, el porcentaje de nulíparas en celo con el tratamiento "b".
En R podemos hacerlo muy rápido. Tenemos que cargar la librería *emmeans* escribiendo en la consola:

```
library(emmeans),
```

y con la instrucción:

```
emmeans(model3, pairwise~as.factor(tratamiento), type
="response")
```

obtendremos:

```
> emmeans(model3, pairwise~as.factor(tratamiento), type
="response")

$emmeans
 Tratamiento  prob   SE     df  asymp.LCL  asymp.UCL
 a            0.375  0.0359  Inf  0.307      0.447
 b            0.633  0.0340  Inf  0.564      0.697

Confidence level used: 0.95
Intervals are back-transformed from the logit scale
```

Que, como vemos, coincide con nuestros resultados. Además, tenemos los intervalos de confianza de la probabilidad de éxito de cada tratamiento. En el caso del tratamiento "a" la probabilidad de éxito va de 30,7 a 44,7 %.
También esta última instrucción nos da el valor de la *odds ratio*.

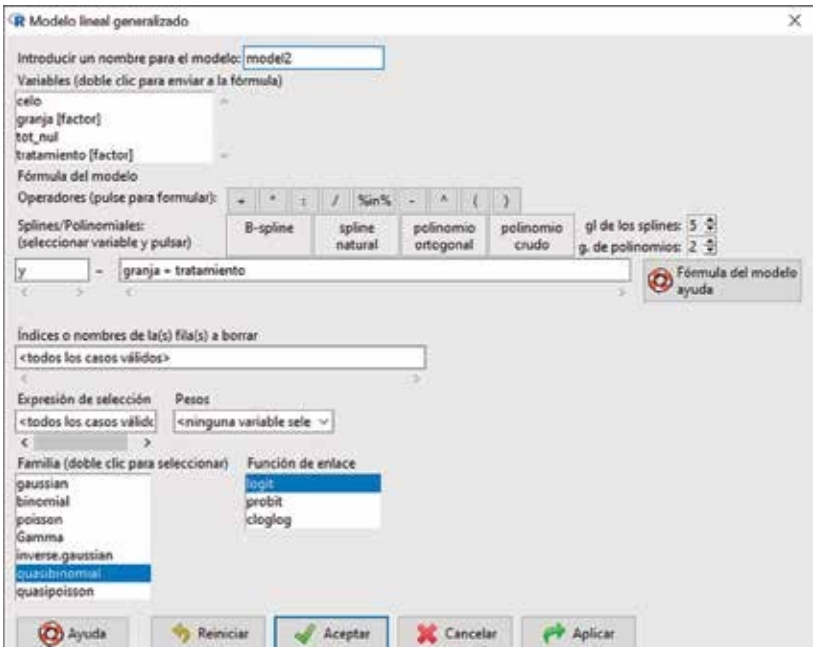

Figura 2. Modelo sin interacción.

```
$contrasts
 Contrast  odds.ratio  SE      df  null  z.ratio  p.value
 a/b       0.347       0.0736  Inf  1     -4.992   <.0001
```

Recordemos primero que una *odds ratio*, en castellano se traduce como razón de probabilidades (también razón de posibilidades). La *odds* de un evento sería la probabilidad de que ocurra un evento dividido por la probabilidad de que no ocurra.
En el grupo "a" la *odds* del evento celo es 0,347 / (1 - 0,347) = 0,653, que se interpretaría como cuán probable es que el celo ocurra con relación a que no ocurra. En cambio en el grupo "b" es 0,633/(1-0,633), 1,49 dando a entender que es 1,5 veces más probable que haya celo que no en el grupo "b".
Así, el *odds ratio* es el resultado de dividir el *odds* de una categoría entre el *odds* de la otra. El concepto de *odds ratio* se puede ver también como la "ventaja" utilizada habitualmente en las apuestas. Es decir, qué ventaja tiene utilizar un tratamiento frente a utilizar otro. Por ejemplo, un *odds* igual a 2 equivaldría a decir que la "ganancia" con el tratamiento en estudio será el doble.
En nuestro caso, R nos informa que la *odds ratio* a/b es igual a 0,347 pero sería más fácil interpretarlo al revés, dividiendo 1 entre 0,347, que es igual a 2,88, con lo que podríamos decir que el tratamiento "b" da lugar a la salida en celo de las nulíparas 2,88 veces más que el tratamiento "a", siendo esta *odds ratio*, estadísticamente significativa ($p < 0.001$).