APLICACIÓN DE LA REGRESIÓN LOGÍSTICA:

# ESTUDIO DEL HÁBITAT DEL *TRITÓN CRESTADO GIGANTE* (*TRITURUS CRISTATUS*, LAURENTI, 1768)

Autoría de la foto

## 1. Introducción.

El tritón crestado gigante es una especie de anfibio salamándrido común en Gran Bretaña y el norte de Europa. Se caracteriza por un pliegue de piel en forma de cresta presente en machos durante época reproductiva que recorre desde su nuca hasta el final de la cola. (Wikipedia).

## 2. Datos del estudio.

El archivo newt-regression.rdata (extraído de dataanalytics.org.uk) contiene datos procedentes de un estudio sobre esta especie realizado en la comarca de Buckinghamshire, situada al sur de Inglaterra entre Oxford y Londres. Para la realización del estudio se seleccionaron al azar 200 charcas de esta zona. En cada charca se identificó la presencia o no de tritones y se evaluó una serie de variables que describian las características de la charca.

Al leer este archivo en R, se carga en el entorno de trabajo el data.frame `gcn` cuya estructura es la siguiente:

```
load("newt-regression.rdata")
head(gcn)
```

```
  presence area dry water shade bird fish other.ponds land macro  HSI
1        0  100   1     2    10    2    1           2    1    30 0.53
2        0   40   1     2    90    1    3           1    1     0 0.47
3        0  150   1     3    20    2    3           4    1    80 0.69
4        0   25   1     2    90    1    3           4    1     0 0.51
5        1  800   1     2    90    1    4           5    3     0 0.70
6        0  275   1     3     5    2    1           2    4    50 0.72
```

Cada fila del data.frame corresponde a una charca. Las variables registradas son:

- `presence`: indica la presencia o ausencia de tritones en la charca como variable binaria (0 = ausencia, 1= presencia).
- `area`: superficie del estanque en metros cuadrados.
- `dry`: estacionalidad de la charca; esta variable toma valores de 1 a 4 (de "poco estacional" a "muy estacional"). En este contexto una charca "poco estacional" es una charca que siempre tiene agua, mientras que una charca muy estacional (`dry=4`) sería una charca que se seca varias veces por año.
- `water`: calidad del agua de 1 a 4 siendo el 1 agua de mala calidad y el 4 agua de calidad excelente.
- `shade` grado de sombra a que se encuentra la charca, en %.
- `bird`: presencia de aves acuáticas (1-3, 1 = ausentes).
- `fish`: presencia de peces (1-4, 4 = ausentes).

- `other.ponds`: Número de otras charcas en un radio de 1 km.
- `land`: calidad del hábitat terrestre (1-4, 4 = buena).
- `macro`: cobertura de macrófitos (plantas acuáticas) en %.
- `HSI`: índice de idoneidad del hábitat (0-1). Se trata de una medida estándar compilada a partir de 9 medidas descriptivas del hábitat, entre las que se incluyen algunas de las anteriores. El HSI se utiliza para facilitar la evaluación del potencial de las masas de agua para albergar poblaciones de tritón crestado y para dar una medida de reproducibilidad a los estudios.

## 3. Objetivos del estudio.

La tarea de investigación tiene dos objetivos principales:

- Identificar los factores del hábitat más importantes (y significativos) que afectan la presencia o ausencia del tritón crestado.
- Comparar la capacidad predictiva de un modelo basado en factores del hábitat con uno que utilice únicamente el HSI.

El primer objetivo implicará construir un modelo de regresión logística, el cual determinará y mostrará los factores del hábitat que son "más influyentes".

El segundo objetivo será elaborar otro modelo de regresión logística más simple utilizando la variable HSI como único predictor. Posteriormente, se compararán el modelo de múltiples factores y el modelo de HSI para evaluar si existe alguna diferencia sustancial en la capacidad predictiva de ambos modelos.

## 4. Ajuste de la regresión logística usando los factores del hábitat.

```
mdl1 <- glm(presence~area+dry+water+shade+bird+fish+other.ponds+land+macro,
            data=gcn, family=binomial)
summary(mdl1)
```

```
Call:
glm(formula = presence ~ area + dry + water + shade + bird +
    fish + other.ponds + land + macro, family = binomial, data = gcn)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.4494384  1.3142358  -3.386  0.00071 ***
area        -0.0002640  0.0002775  -0.951  0.34148
dry          0.0302195  0.1900303   0.159  0.87365
water        0.4481827  0.2291810   1.956  0.05051 .
shade       -0.0156138  0.0054399  -2.870  0.00410 **
bird         0.2120024  0.3051850   0.695  0.48726
fish         0.5257295  0.2160471   2.433  0.01496 *
other.ponds  0.2480152  0.0815396   3.042  0.00235 **
land         0.0542667  0.1699490   0.319  0.74949
macro        0.0152289  0.0066753   2.281  0.02253 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 261.37  on 199  degrees of freedom
Residual deviance: 216.84  on 190  degrees of freedom
AIC: 236.84

Number of Fisher Scoring iterations: 4
```

Podemos utilizar un método paso a paso para seleccionar el modelo con menor AIC:

```
mdl0 <- glm(presence~1,data=gcn,family=binomial)
mdl1.both <- step(mdl0,direction="both",scope=list(lower=mdl0, upper=mdl1),trace=0)
mdl1.forw <- step(mdl0,direction="forward",scope=list(lower=mdl0, upper=mdl1),trace=0)
mdl1.back <- step(mdl1,direction="backward",scope=list(lower=mdl0, upper=mdl1),trace=0
AIC(mdl1.both, mdl1.forw, mdl1.back)
```

```
          df      AIC
```

```
mdl1.both  6 230.5233
mdl1.forw  6 230.5233
mdl1.back  6 230.5233
```

Como vemos, da igual qué método utilicemos, el modelo seleccionado es el mismo en los tres casos. Este modelo es el siguiente:

```
summary(mdl1.both)
```

```
Call:
glm(formula = presence ~ macro + other.ponds + fish + shade +
    water, family = binomial, data = gcn)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.130592   0.965606  -4.278 1.89e-05 ***
macro        0.016121   0.006518   2.473  0.01339 *
other.ponds  0.238078   0.079631   2.990  0.00279 **
fish         0.562670   0.178477   3.153  0.00162 **
shade       -0.015162   0.005372  -2.822  0.00477 **
water        0.446487   0.224915   1.985  0.04713 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 261.37  on 199  degrees of freedom
Residual deviance: 218.52  on 194  degrees of freedom
AIC: 230.52

Number of Fisher Scoring iterations: 4
```

Las variables explicativas seleccionadas son `macro`, `other.ponds`, `fish`, `shade` y `water`.

## 5. Deviance

Hasta ahora en los modelos lineales al final del `summary` nos aparecía el error estándar residual, el $R^2$ y el $R^2$ ajustado. Sin embargo en regresión logística nos aparecen la *null deviance* y la *residual deviance* ¿Qué representan estos valores?

Para entender el significado de la *deviance* es preciso entender el procedimiento de estimación de los parámetros de la regresión logística. El método seguido es el de máxima verosimilitud. Recordemos que el modelo de regresión logística es de la forma:

$$P\left(Y=1\left|X_1=x_1,\dots,X_p=x_p\right.\right)=\frac{e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}{1+e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}$$

Nótese que este modelo implica que:

$$P\left(Y=0\left|X_1=x_1,\dots,X_p=x_p\right.\right)=1-\frac{e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}{1+e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}$$

Dada una observación arbitraria $y$ de la variable respuesta ($y$ puede ser, por tanto 0 ó 1) las dos expresiones anteriores pueden combinarse mediante:

$$P\left(Y=y\left|X_1=x_1,\dots,X_p=x_p\right.\right)=\left(\frac{e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}{1+e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}\right)^y\left(1-\frac{e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}{1+e^{\beta_0+\beta_1x_1+\cdots+\beta_px_p}}\right)^{1-y}$$

Los datos de la muestra son de la forma:

$$\begin{pmatrix} y_1 & x_{11} & x_{21} & \dots & x_{p1} \\ y_2 & x_{12} & x_{22} & \dots & x_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_n & x_{1n} & x_{2n} & \dots & x_{pn} \end{pmatrix}$$

donde los $y_i$ son los valores (0 ó 1) de la variable respuesta y $x_{i,j}$ es la observación $j$-ésima de la variable $X_i$.

Si el modelo de regresión logística es adecuado para estos datos, entonces la verosimilitud (probabilidad conjunta) de las observaciones de la variable respuesta es:

$$P\left(Y_1=y_1,Y_2=y_2,\dots,Y_n=y_n\right)=$$
$$=\prod_{i=1}^{n}\left(\frac{e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}{1+e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}\right)^{y_i}\left(1-\frac{e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}{1+e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}\right)^{1-y_i}$$

y la log-verosimilitud:

$$\ell\left(y_1,\dots,y_n\right)=\log\left(P\left(Y_1=y_1,Y_2=y_2,\dots,Y_n=y_n\right)\right)=$$
$$=\sum_{i=1}^{n}\left[y_i\log\left(\frac{e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}{1+e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}\right)+(1-y_i)\log\left(1-\frac{e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}{1+e^{\beta_0+\beta_1x_{1i}+\cdots+\beta_px_{pi}}}\right)\right]$$

Los estimadores $\hat{\beta}_i$ son los valores de $\beta_i$ que maximizan esta verosimilitud y se obtienen numéricamente mediante algoritmos como el de *Newton-Raphson*. Llamaremos $\ell\left(\hat{\beta}\right)$ al valor de la log-verosimilitud, que se alcanza con los $\hat{\beta}_i$ estimados para el modelo, y que corresponde a la máxima verosimilitud posible con los datos disponibles.

En nuestro modelo para la presencia de tritones, la log-verosimilitud $\ell\left(\hat{\beta}\right)$ del modelo finalmente ajustado puede calcularse como:

```
p <- predict(mdl1.both,type="response")
y <- gcn$presence
loglik <- sum(y*log(p)+(1-y)*log(1-p))
loglik
```

```
[1] -109.2617
```

Este valor *per se* no proporciona información respecto a si el modelo presenta un buen ajuste a los datos (que sea el *mejor* modelo que hayamos podido ajustar no significa que sea necesariamente un *buen* modelo para los datos disponibles). Una idea para valorar el ajuste conseguido podría ser comparar la verosimilitud alcanzada por este modelo con la verosimilitud que tendría un modelo que fuera capaz de predecir *exactamente* los valores de la variable respuesta $Y$. La verosimilitud de ese modelo ideal (que suele recibir el nombre de *modelo saturado*) sería, obviamente:

$$P\left(Y_1=y_1,Y_2=y_2,\dots,Y_n=y_n\right)=1$$

La log-verosimilitud del modelo saturado sería el logaritmo de esta probabilidad y por tanto su valor es $\ell\left(\text{Modelo Saturado}\right)=0$. Por tanto, la diferencia entre la log-verosimilitud de un modelo (ideal) que sea capaz de predecir exactamente los valores de la variable respuesta y la verosimilitud del modelo que hemos ajustado es:

$$\ell\left(\text{Modelo Saturado}\right)-\ell\left(\hat{\beta}\right)=-\ell\left(\hat{\beta}\right)$$

Si tenemos en cuenta que la verosimilitud produce siempre valores menores que uno, su logaritmo (la log-verosimilitud) será siempre un número negativo. Por tanto en la expresión anterior la diferencia entre la log-verosimilitud de modelo saturado y la del modelo ajustado es siempre un valor positivo, que podría llegar a ser cero si el modelo ajustado fuese capaz de hacer predicciones exactas.

Por tanto, la intuición nos indica que cuanto mayor sea el valor de $-\ell\left(\widehat{\beta}\right)$, más lejos estará nuestro modelo de hacer predicciones exactas. Asimismo, si el valor de $-\ell\left(\widehat{\beta}\right)$ fuera pequeño, ello sería una indicación de que el modelo presenta un buen ajuste. ¿Cómo podemos juzgar si el valor de $-\ell\left(\widehat{\beta}\right)$ es grande o pequeño? La respuesta nos la proporciona el **test de razón de verosimilitudes**. En esencia, dados dos modelos $M_1$ y $M_2$ que expliquen la relación entre una variable respuesta $Y$ y unas variables explicativas $X_1, ..., X_p$, si $V(M_1)$ y $V(M_2)$ son, respectivamente, las máximas verosimilitudes alcanzadas por cada uno de estos modelos, el cociente $\frac{V(M_1)}{V(M_2)}$ será un número próximo a 1 si ambos modelos son equivalentes y distinto de uno si uno de los modelos es mejor que otro. El *teorema de Wilks* indica que, cuando $n$ (el número de observaciones) es grande y no hay diferencias entre ambos modelos más allá de las debidas al puro azar, la variable aleatoria:

$$\Delta = 2\log\left(\frac{V\left(M_1\right)}{V\left(M_2\right)}\right) = 2\left(\log\left(V(M_1) - \log\left(V\left(M_2\right)\right)\right)\right) = 2\left(\ell\left(M_1\right) - \ell\left(M_2\right)\right)$$

sigue aproximadamente una distribución chi-cuadrado con $\nu_1 - \nu_2$ grados de libertad, siendo $\nu_i$ el número de parámetros de cada modelo. El valor de $\Delta$ calculado sobre una muestra de observaciones se denomina *Deviance*, y como vemos es la diferencia de log-verosimilitudes multiplicada por 2.

Si volvemos a nuestro ejemplo, si multiplicamos por 2 la diferencia entre las log-verosimilitudes del modelo saturado y el modelo que hemos ajustado obtenemos la *deviance*:

$$D = 2 \cdot \left(\ell\left(\text{Modelo Saturado}\right) - \ell\left(\widehat{\beta}\right)\right) = -2\ell\left(\widehat{\beta}\right)$$

Esta deviance recibe el nombre de *deviance residual* (pues de alguna forma mide la diferencia entre nuestro modelo y un modelo que se ajuste exactamente a los datos). Si nuestro modelo predijera exactamente los valores observados de $Y$ su deviance sería 0; esta deviance sería tanto mayor cuanto peores sean las predicciones.

En nuestro ejemplo la deviance residual es:

```
residual_Deviance=-2*loglik
residual_Deviance
```

```
[1] 218.5233
```

que coincide con el valor mostrado por R al hacer `summary` de nuestro modelo. Dado que $n$ es grande (se han observado n=200 tritones), si entre el modelo saturado y el modelo ajustado no hay diferencias significativas, el valor de esta deviance sigue una distribución chi-cuadrado con 200-6=194 grados de libertad (pues el modelo saturado, para hacer predicciones exactas, requiere 200 parámetros[1], mientras que nuestro modelo usa 6 parámetros). La probabilidad de que una variable chi cuadrado con 194 grados de libertad llegue a alcanzar un valor tan grande como el observado (218.5233) es:

```
1-pchisq(218.5233, 194)
```

```
[1] 0.1094251
```

Este valor ($> 0.05$) indica que no hay diferencias significativas entre el modelo saturado (que hace predicciones exactas) y el modelo que hemos ajustado, lo que significa que estamos ante un buen modelo. El test de razón de verosimilitudes permite hacer un contraste adicional para decidir si, globalmente, las variables incluidas en el modelo tienen efecto significativo sobre la respuesta; para ello deberemos comparar la verosimilitud del modelo que hemos ajustado con la verosimilitud de un modelo que solo tenga término independiente (o *modelo nulo*, modelo sin predictores), esto es:

$$P\left(Y=y\right)=\left(\frac{e^{\beta_0}}{1+e^{\beta_0}}\right)^{y}\left(1-\frac{e^{\beta_0}}{1+e^{\beta_0}}\right)^{1-y}$$

Llamando $p=\frac{e^{\beta_0}}{1+e^{\beta_0}}$, este modelo significa que la probabilidad de que haya tritones en una charca arbitraria es $p$ siempre, independientemente del valor del resto de variables. Este valor de $p$ puede estimarse simplemente como el número de charcas en que se han detectado tritones dividido entre el total de charcas:

```
k <- sum(gcn$presence)
n <- nrow(gcn)
p <- k/n
```

[1] En realidad, hacer predicciones exactas sería equivalente a disponer *a priori* de las probabilidades $p_i$ de existencia de tritones en cada charca $i=1,\ldots,200$, de forma que $p_i=1$ si en dicha charca hay tritones y $p_i=0$ si no los hay

La log-verosimilitud de este modelo sería entonces:

```
loglik0 <- k*log(p)+(n-k)*log(1-p)
```

y su deviance (dos veces la diferencia con el modelo saturado) recibe el nombre de *null deviance*:

```
null_Deviance <- -2*loglik0
null_Deviance
```

```
[1] 261.3673
```

Si revisamos el `summary` de nuestro modelo vemos que R nos muestra también el valor de la deviance del modelo nulo.

La deviance del modelo nulo es siempre mayor que la deviance residual del modelo ajustado. El coeficiente $R_L^2$ de Cohen (a veces llamado pseudo-$R^2$) se define como:

$$R_L^2 = \frac{D_{\text{null}} - D_{\text{residual}}}{D_{\text{null}}} = 1 - \frac{D_{\text{residual}}}{D_{\text{null}}}$$

y mide la reducción proporcional en la deviance del modelo nulo que se consigue al añadir nuevas variables; es un concepto parecido, pero no idéntico, al coeficiente de determinación $R^2$ que se usa en el modelo lineal; en el caso del modelo lineal, la $R^2$ mide en qué proporción se reduce la variabilidad en las predicciones de la variable respuesta cuando se usan las variables predictoras. En el caso de la regresión logística, la $R_L^2$ mide en qué proporción se reduce la *incertidumbre* sobre las predicciones al utilizar las variables predictoras.

En nuestro modelo para la presencia de tritones la pseudo-$R^2$ puede calcularse como:

```
1-mdl1.both$deviance/mdl1.both$null.deviance
```

```
[1] 0.1639223
```

Para decidir si este incremento en pseudo-$R^2$ es significativo (aunque sea pequeño), podemos hacer también el test de razón de verosimilitudes comparando el modelo nulo con el modelo ajustado:

```
with(mdl1.both,1-pchisq(null.deviance-deviance,df.null-df.residual))
```

```
[1] 3.974217e-08
```

Este valor de $p$ es muy pequeño, mucho menor que el nivel de significación estándar del 5%, por lo cuál podemos asegurar que modelo ajustado con las cinco variables elegidas predice la variable respuesta significativamente mejor que el modelo sin predictores. Este test podría realizarse en R ajustando directamente el modelo nulo:

```
mdl0 <- glm(presence~1,data=gcn,family=binomial)
```

y comparándolo con el modelo anterior usando `anova` y especificando `test="Chisq"`:

```
anova(mdl0,mdl1.both,test="Chisq")
```

```
Analysis of Deviance Table

Model 1: presence ~ 1
Model 2: presence ~ macro + other.ponds + fish + shade + water
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       199     261.37
2       194     218.52  5   42.844 3.974e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En general puede usarse la función `anova` en R para comparar modelos anidados[2] que se han ajustado mediante regresión logística, utilizando siempre la opción `test="Chisq"`.

Otra opción para realizar este test es utilizar la función `lrtest` del paquete `lmtest`:

```
library(lmtest)
lrtest(mdl0,mdl1.both)
```

```
Likelihood ratio test

Model 1: presence ~ 1
Model 2: presence ~ macro + other.ponds + fish + shade + water
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   1 -130.68
2   6 -109.26  5 42.844  3.974e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[2]Un modelo está anidado en otro cuando las variables del primero forman también parte del segundo

## 6. Presentación e interpretación de los resultados en términos de Odds-Ratio

Podemos calcular la OR entre la presencia de tritones y cada una de las variables del modelo:

```
# Extraigo la tabla de estimadores y p-valores del modelo:
orp <- summary(mdl1.both)$coefficients[-1,] %>% # Elimino el término independiente
  data.frame() %>%
  rownames_to_column() %>%
  mutate(OR=exp(Estimate)) %>% # Calculo la OR
  rename(P=Pr...z..) %>%
  select(rowname,OR,P)

# Calculo los intervalos de confianza de las OR
orci <- exp(confint(mdl1.both))[-1,] %>% # Elimino el término independiente
  data.frame() %>%
  rownames_to_column() %>%
  mutate(`CI95%`=sprintf("[%.2f, %.2f]",.[[2]],.[[3]])) %>%
  full_join(orp) %>%
  mutate(P=formatPval(P)) %>%
  select(variable=rowname,OR,`CI95%`,P)

# Lo presento en una tabla:
orci %>% knitr::kable() %>%
  kableExtra::kable_classic(full_width=FALSE)
```

| variable | OR | CI95% | P |
|---|---|---|---|
| macro | 1.0162519 | [1.00, 1.03] | 0.0134 |
| other.ponds | 1.2688080 | [1.09, 1.49] | 0.0028 |
| fish | 1.7553534 | [1.25, 2.53] | 0.0016 |
| shade | 0.9849521 | [0.97, 1.00] | 0.0048 |
| water | 1.5628118 | [1.01, 2.45] | 0.0471 |

De esta tabla se sigue que:

- La cobertura de macrófitos (`macro`), la presencia de tritones en charcas cercanas y la calidad del agua favorecen la presencia de tritones (OR mayores que 1 y significativas en todos los casos).
- La variable `fish` se ha codificado en una escala de 1 (abundancia de peces) a 4 (ausencia de peces). La OR de 1.76 indica en este caso que la ausencia de peces favorece la presencia de tritones (probablemente los peces se comen a los tritones en alguna fase del ciclo vital de estos últimos).
- La variable `shade` muestra una OR menor que 1, lo que indica que un mayor porcentaje de sombra inhibe la presencia de tritones.

## 7. Probabilidades predichas por el modelo

El efecto de estas variables sobre la presencia de tritones se puede ilustrar mostrando gráficamente como varía la probabilidad de encontrar tritones a medida que cambian los valores de esas variables.

Primero evaluamos el valor medio de cada variable predictora:

```
media <- gcn %>%
  summarize(across(c(macro,other.ponds,fish,shade,water),mean))
media %>% knitr::kable() %>% kableExtra::kable_classic(full_width=FALSE)
```

| macro | other.ponds | fish | shade | water |
|---|---|---|---|---|
| 23.925 | 3.445 | 3 | 38.365 | 2.44 |

Y también sus valores mínimo y máximo:

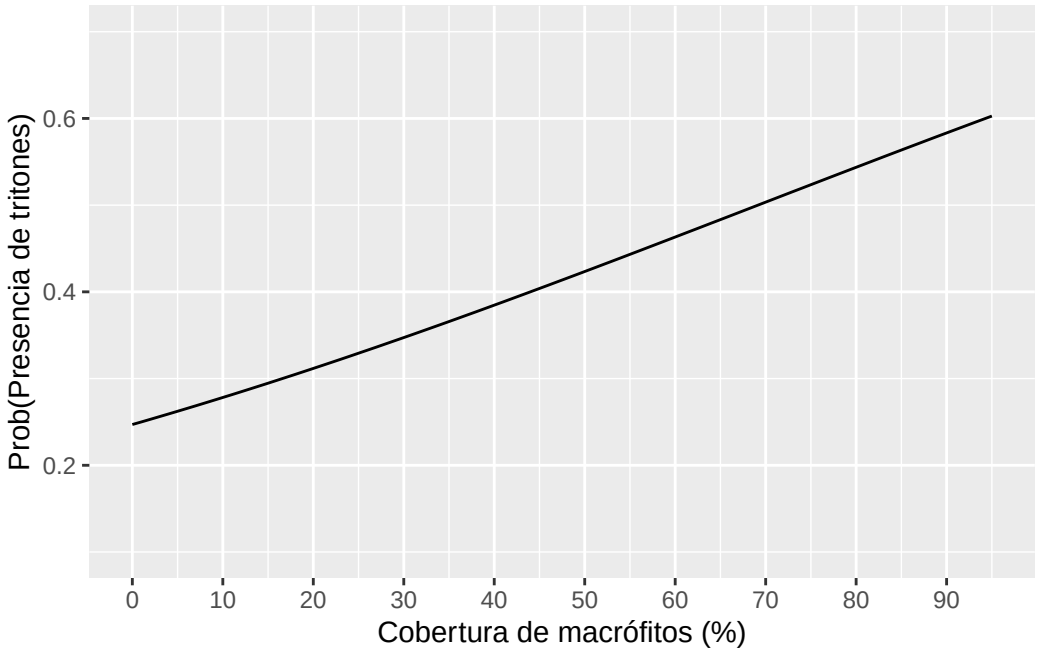
```
rango <- gcn %>%
  summarize(across(c(macro,other.ponds,fish,shade,water),range))
rango %>% knitr::kable() %>% kableExtra::kable_classic(full_width=FALSE)
```

| macro | other.ponds | fish | shade | water |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |

| | | | | |
|---|---|---|---|---|
| 95 | 9 | 4 | 100 | 4 |

*Efecto de la cobertura de macrófitos.* Calculamos las probabilidades predichas por el modelo para la presencia de tritones a medida que cambia la cobertura de macrófitos, en el supuesto de que el resto de variables tomen un valor constante (su media):

```
newdf <- data.frame(macro=0:95,
                    other.ponds=media$other.ponds,
                    fish=media$fish,
                    shade=media$shade,
                    water=media$water)
newdf %>%
  mutate(p=predict(mdl1.both,newdata = newdf,type="response")) %>%
  ggplot(aes(x=macro,y=p)) +
  geom_line()+
  ylim(0.1,0.7)+
  scale_x_continuous(breaks=seq(0,90,by=10))+
  labs(x="Cobertura de macrófitos (%)",y="Prob(Presencia de tritones)")
```




Esta gráfica nos indica que cuando las variables `other.ponds`, `fish`, `water` y `shade` están en sus valores medios, la probabilidad de detectar tritones cuando no hay macrófitos

es aproximadamente del 25%; a medida que aumenta el porcentaje de macrófitos, se incrementa también la probabilidad de detectar tritones, que llega a ser aproximadamente del 60% cuando la cobertura de macrófitos es del 95%.

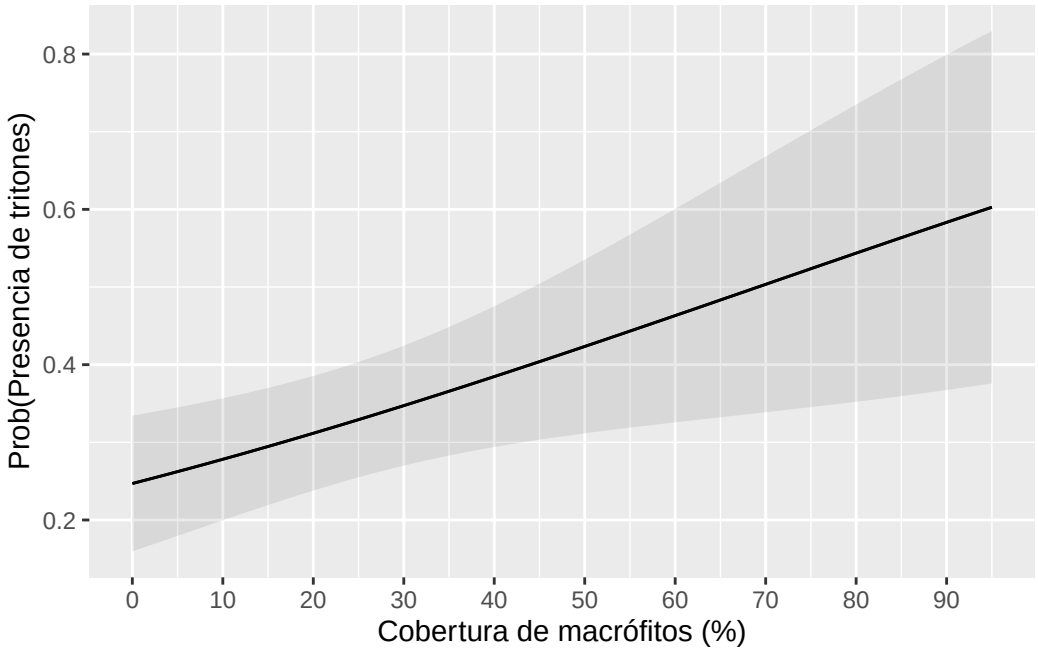
El paquete `marginaleffects` permite además calcular intervalos de confianza para las probabilidades predichas:

```
library(marginaleffects)
predictions(mdl1.both, newdata=newdf, type="response",by="macro") %>%
  mutate(`CI95%`=sprintf("[%.2f; %.2f]",conf.low,conf.high)) %>%
  select(macro,other.ponds:water,prob=estimate,`CI95%`) %>%
  tibble() %>%
  head() %>%  # Se muestran solo los primeros valores de la tabla
  knitr::kable() %>% kableExtra::kable_classic(full_width=FALSE)
```

| macro | other.ponds | fish | shade | water | prob | CI95% |
|---|---|---|---|---|---|---|
| 0 | 3.445 | 3 | 38.365 | 2.44 | 0.2470000 | [0.16; 0.33] |
| 1 | 3.445 | 3 | 38.365 | 2.44 | 0.2500106 | [0.16; 0.34] |
| 2 | 3.445 | 3 | 38.365 | 2.44 | 0.2530456 | [0.17; 0.34] |
| 3 | 3.445 | 3 | 38.365 | 2.44 | 0.2561049 | [0.17; 0.34] |
| 4 | 3.445 | 3 | 38.365 | 2.44 | 0.2591883 | [0.18; 0.34] |
| 5 | 3.445 | 3 | 38.365 | 2.44 | 0.2622957 | [0.18; 0.35] |

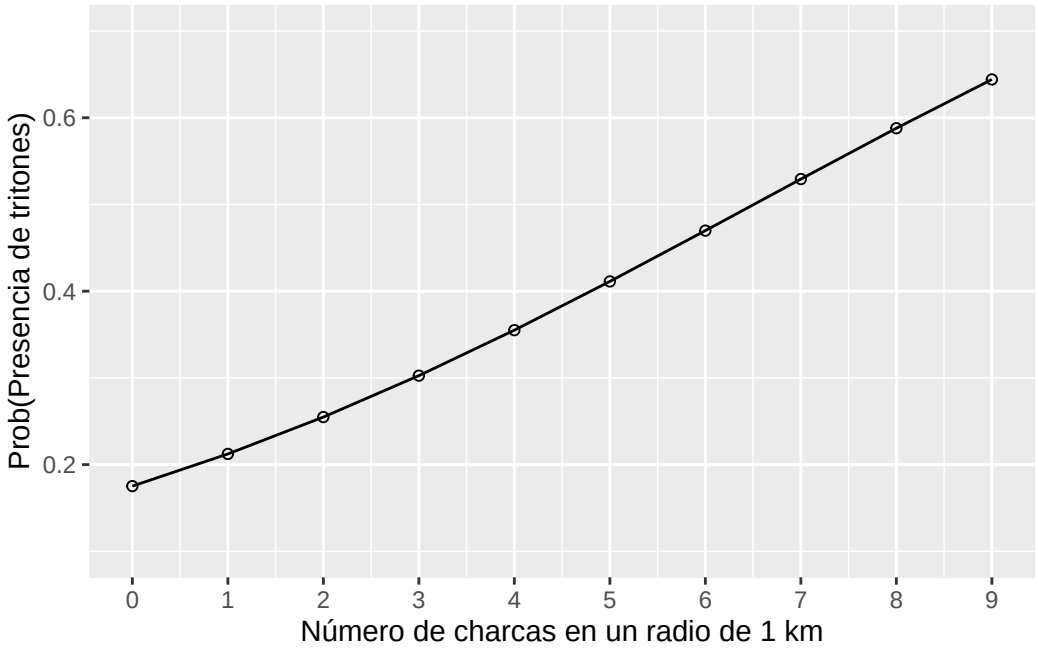
Gráficamente:

```
plot_predictions(mdl1.both, newdata=newdf, type="response",by="macro") +
  scale_x_continuous(breaks=seq(0,90,by=10))+
  labs(x="Cobertura de macrófitos (%)",y="Prob(Presencia de tritones)")
```




*Efecto de la presencia de tritones en charcas cercanas.*

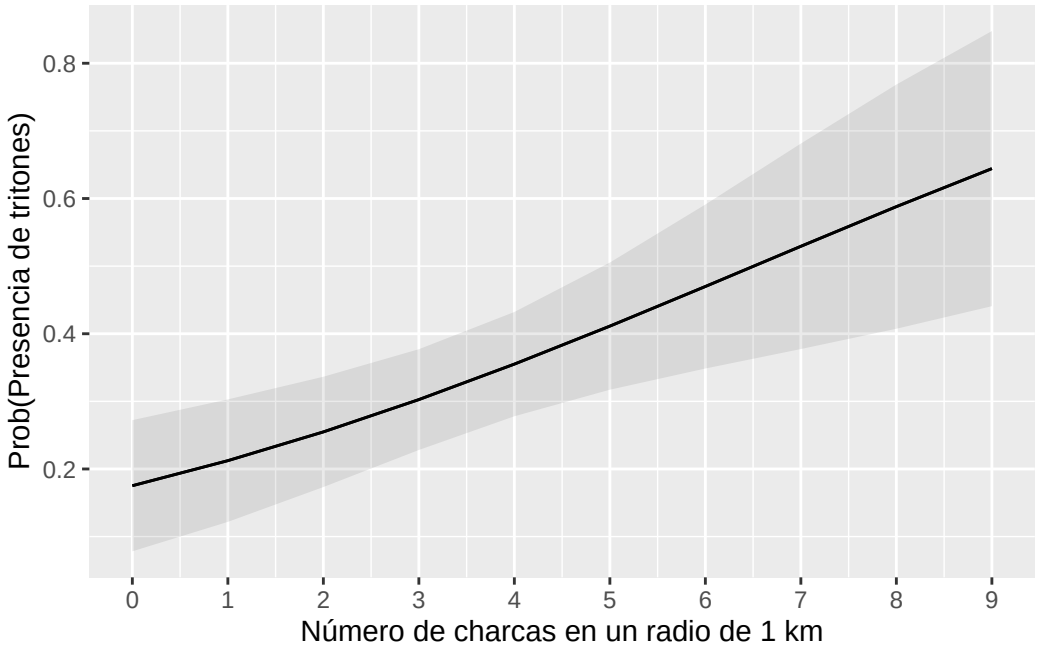
```
newdf <- data.frame(macro=media$macro,
                    other.ponds=0:9,
                    fish=media$fish,
                    shade=media$shade,
                    water=media$water)
newdf %>%
  mutate(p=predict(mdl1.both,newdata = newdf,type="response")) %>%
  ggplot(aes(x=other.ponds,y=p)) +
  geom_line()+geom_point(shape=21) +
  ylim(0.1,0.7) +
  scale_x_continuous(breaks=0:9)+
  labs(x="Número de charcas en un radio de 1 km",y="Prob(Presencia de tritones)")
```




Al igual que en el caso anterior, cuando las variables `macro`, `fish`, `water` y `shade` están en sus valores medios, la probabilidad de que haya tritones en una charca se incrementa a medida que aumenta el número de charcas vecinas con tritones.

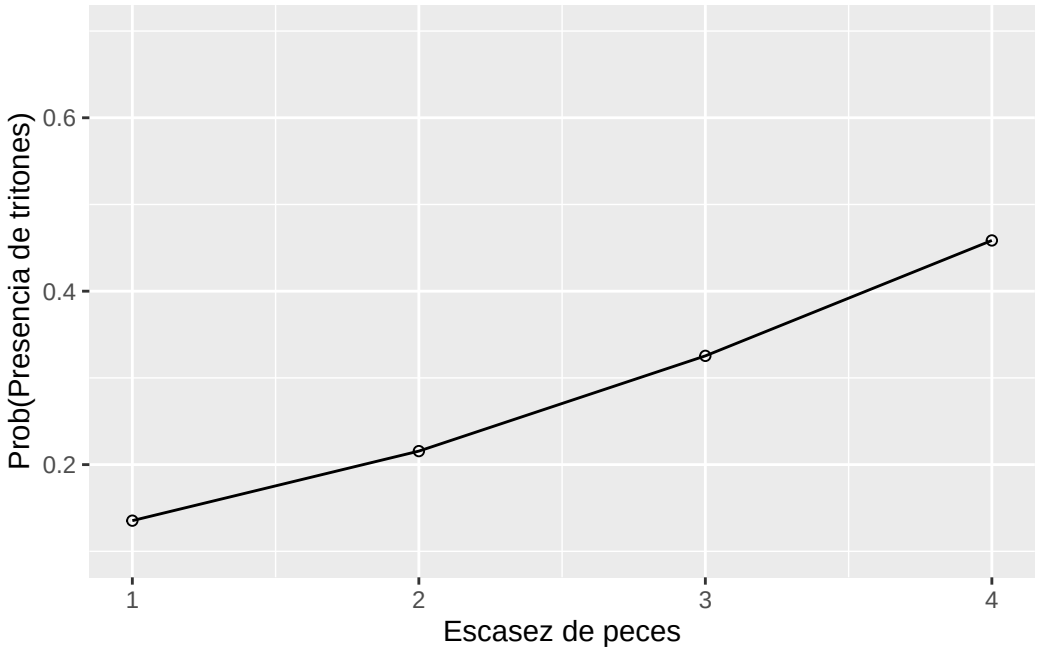
Con el intervalo de confianza:

```
plot_predictions(mdl1.both, newdata=newdf, type="response",by="other.ponds") +
  scale_x_continuous(breaks=0:9)+
  labs(x="Número de charcas en un radio de 1 km",y="Prob(Presencia de tritones)")
```




*Efecto de la ausencia de peces en la charca.*

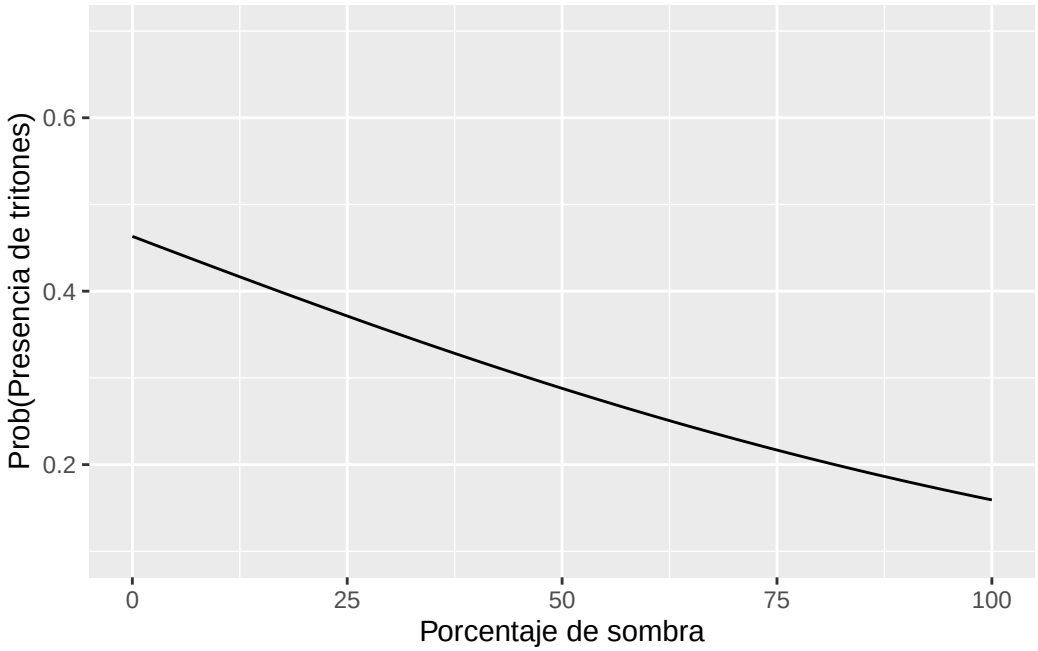
```
newdf <- data.frame(macro=media$macro,
                    other.ponds=media$other.ponds,
                    fish=1:4,
                    shade=media$shade,
                    water=media$water)
newdf %>%
  mutate(p=predict(mdl1.both,newdata = newdf,type="response")) %>%
  ggplot(aes(x=fish,y=p)) +
  geom_line()+geom_point(shape=21) +
  ylim(0.1,0.7) +
  scale_x_continuous(breaks=1:4)+
  labs(x="Escasez de peces",y="Prob(Presencia de tritones)")
```




A medida que aumenta la "escasez" de peces, aumenta la probabilidad de que haya tritones en la charca (manteniendo constantes todas las demás variables)

*Efecto del porcentaje de sombra.*

```
newdf <- data.frame(macro=media$macro,
                    other.ponds=media$other.ponds,
                    fish=media$fish,
                    shade=0:100,
                    water=media$water)
newdf %>%
  mutate(p=predict(mdl1.both,newdata = newdf,type="response")) %>%
  ggplot(aes(x=shade,y=p)) +
  geom_line()+
  ylim(0.1,0.7) +
  labs(x="Porcentaje de sombra",y="Prob(Presencia de tritones)")
```

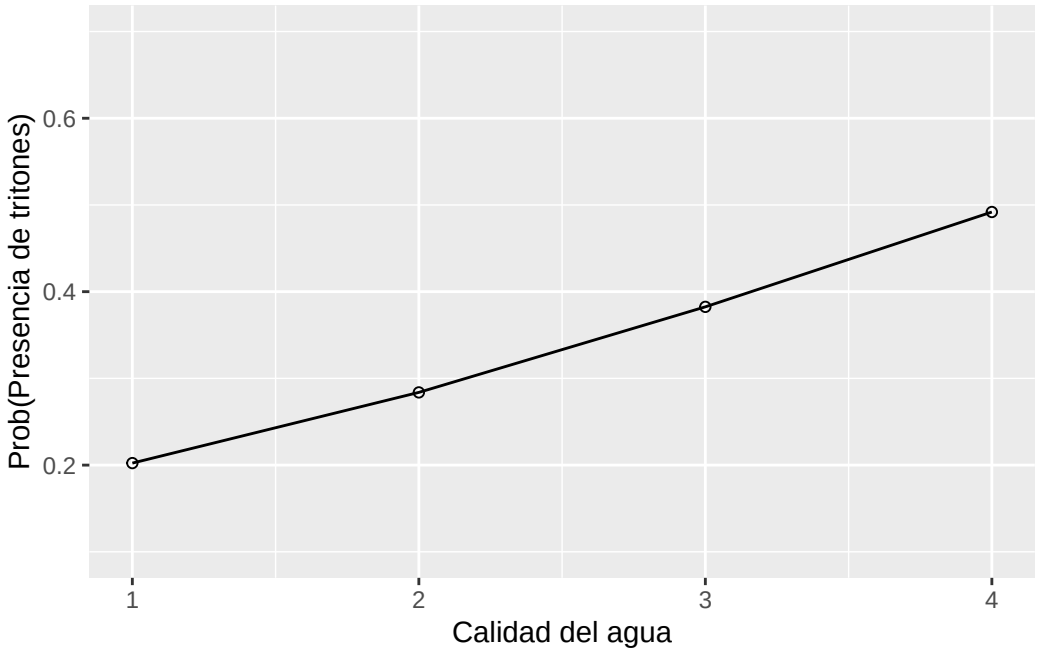



En este caso observamos que a medida que aumenta el porcentaje de sombra en la charca disminuye la probabilidad de encontrar tritones. Esta probabillidad es de aproximadamente de 0.46 cuando no hay sombra, y disminuye hasta aproximadamente 0.16 cuando la charca está totalmente a la sombra.

*Efecto de la calidad del agua.*

```
newdf <- data.frame(macro=media$macro,
                    other.ponds=media$other.ponds,
                    fish=media$fish,
                    shade=media$shade,
                    water=1:4)
newdf %>%
  mutate(p=predict(mdl1.both,newdata = newdf,type="response")) %>%
  ggplot(aes(x=water,y=p)) +
  geom_line()+
  geom_point(shape=21) +
  ylim(0.1,0.7)+
```

```
  scale_x_continuous(breaks=1:4)+
  labs(x="Calidad del agua",y="Prob(Presencia de tritones)")
```

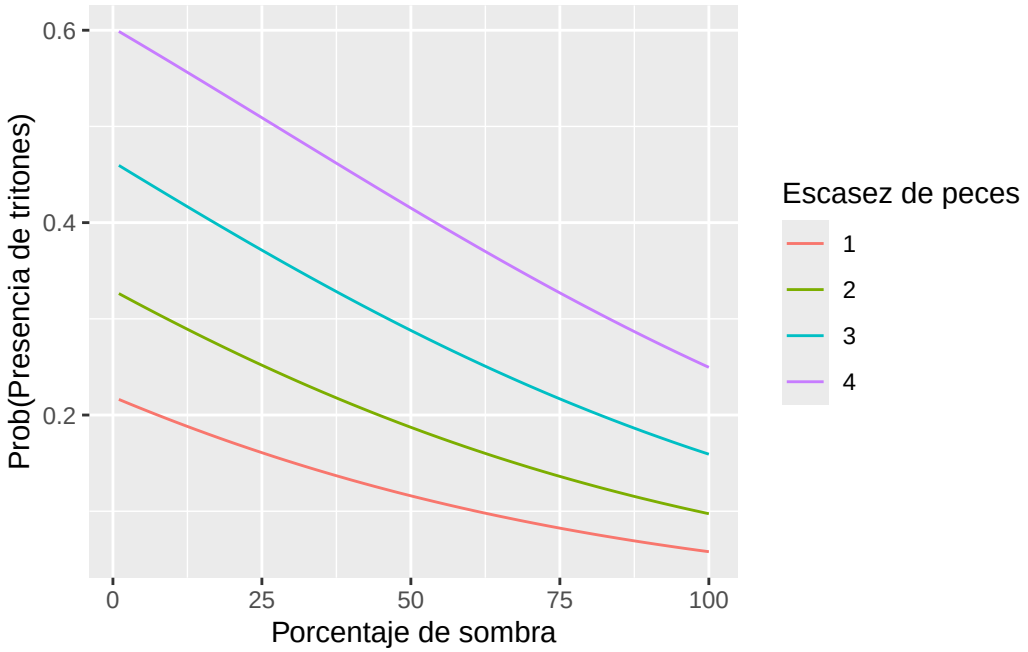



La probabilidad de que haya tritones es más baja en agua de mala calidad y aumenta a medida que aumenta la calidad del agua.

*Efecto combinado de varias variables.* Es posible realizar predicciones con el modelo para diversas combinaciones de valores de las variables, de tal forma que se pueda visualizar como varía la probabilidad a medida que cambian los valores de una variable, teniendo en cuenta además los valores de otra. Por ejemplo, en la siguiente gráfica vemos como cambia la probabilidad de que haya tritones en función del porcentaje de sombra en la charca teniendo en cuenta además la presencia/ausencia de peces:

```
shf <- expand.grid(shade=1:100,fish=1:4)
newdf <- data.frame(shf,
                    macro=media$macro,
                    other.ponds=media$other.ponds,
                    water=media$water)
newdf %>%
  mutate(p=predict(mdl1.both,newdata = newdf,type="response"),
```

```
        fish=factor(fish)) %>%
ggplot(aes(x=shade,y=p,color=fish,group=fish)) +
geom_line()+
labs(x="Porcentaje de sombra",y="Prob(Presencia de tritones)",
     color="Escasez de peces")
```




Así vemos que cuando los peces son abundantes (`fish=1`) y además hay mucha sombra en la charca, la probabilidad de que haya tritones es muy baja (0.06). En las mismas condiciones de sombra, si no hay peces (`fish=4`) la probabilidad de que haya tritones aumenta hasta 0.25. Cuando no hay peces (`fish=4`) y no hay sombra (`shade=0`), la probabilidad de encontrar tritones sube hasta 0.60

## 8. Capacidad predictiva del modelo.

Más allá de decidir si existe o no efecto significativo de cada variable sobre la respuesta (en este caso la presencia/ausencia de tritones en la charca), para valorar el buen o mal ajuste de la regresión logística se suele usar el área bajo la curva ROC (Receiver Operating Curve). Esta curva, en realidad, permite valorar la capacidad predictiva de la regresión logística.

Tengamos en cuenta que el modelo de regresión permite estimar las probabilidades de que la variable respuesta tome los valores 0 ó 1 a partir de las variables explicativas que se han considerado en el modelo. En nuestro ejemplo de los tritones, las probabilidades predichas por el modelo elegido se muestran en la tabla siguiente (solo las 15 primeras filas):

```
gcnp <- gcn %>%
  mutate(prob=predict(mdl1.both,type="response")) %>%
  select(presence,water,shade,fish,other.ponds,macro,prob)
head(gcnp,15)
```

| | presence | water | shade | fish | other.ponds | macro | prob |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 10 | 1 | 2 | 30 | 0.13391510 |
| 2 | 0 | 2 | 90 | 3 | 1 | 0 | 0.06439633 |
| 3 | 0 | 3 | 20 | 3 | 4 | 80 | 0.69754784 |
| 4 | 0 | 2 | 90 | 3 | 4 | 0 | 0.12326138 |
| 5 | 1 | 2 | 90 | 4 | 5 | 0 | 0.23845771 |
| 6 | 0 | 3 | 5 | 1 | 2 | 50 | 0.26462770 |
| 7 | 0 | 3 | 5 | 1 | 5 | 5 | 0.26244913 |
| 8 | 0 | 2 | 0 | 3 | 4 | 0 | 0.35496232 |
| 9 | 0 | 2 | 95 | 4 | 0 | 5 | 0.08732461 |
| 10 | 1 | 3 | 15 | 3 | 1 | 60 | 0.46874077 |
| 11 | 0 | 2 | 100 | 3 | 3 | 0 | 0.08693849 |
| 12 | 0 | 2 | 70 | 4 | 3 | 60 | 0.40931785 |
| 13 | 0 | 3 | 10 | 2 | 0 | 40 | 0.23639205 |
| 14 | 0 | 2 | 80 | 3 | 1 | 5 | 0.07988533 |
| 15 | 0 | 2 | 75 | 3 | 1 | 5 | 0.08563821 |

Una vez estimadas estas probabilidades podemos construir una regla de decisión para clasificar las observaciones. Esta regla de clasificación consiste en fijar un valor umbral (*cutoff*) $C$ tal que si la probabilidad predicha es menor o igual que él ($p \leq C$) decidiríamos que en es charca no va a haber tritones; y en caso de ser mayor ($p > C$) decidiríamos que sí.

A modo de ejemplo, si elegimos como *cutoff* el valor $C = 0.5$ (esto es, si la probabilidad predicha es menor que 0.5 decidimos que no hay tritones y si es mayor decidimos que sí), para las dos primeras charcas de la tabla anterior decidiríamos que no hay tritones, para la tercera que sí, etc. Ahora bien, si nos fijamos nos daremos cuenta de que en

la tercera charca (prob=0.697) decidiríamos que sí hay tritones, cuando en realidad no los hubo; en la quinta charca (prob=0.238) decidiríamos que no hay tritones, cuando en realidad sí que los hubo. En definitiva, una vez fijado un *cutoff* habrá casos en que cometeremos el error de decidir 1 cuando la respuesta original fue 0 (esto es un falso positivo) y habrá casos en que cometeremos el error de decidir 0 cuando la respuesta original fue 1 (falso negativo). Si cambiamos el valor de cutoff cambiará también el número de falsos positivos y falsos negativos producidos por el modelo.

Cuando se fija una regla de clasificación como la que hemos definido, se denomina *sensibilidad* a la proporción de verdaderos positivos identificados por la regla; y se denomina *especificidad* a la proporción de verdaderos negativos. Así, para nuestro modelo, si $C = 0.5$:

```
C <- 0.5
gcnp %>%
  mutate(clasif=ifelse(prob<=C,0,1)) %>%
  tabyl(presence,clasif) %>%
  adorn_totals("col") %>%
  adorn_title(col_name="Clasificación")
```

```
          Clasificación
 presence             0  1 Total
        0           110 18   128
        1            37 35    72
```

Por tanto nuestra regla de clasificación identifica correctamente 35 de las 72 charcas donde había tritones (por tanto tiene una sensibilidad de 35/72=0.486). Asimismo identifica como charcas sin tritones 110 de las 128 charcas donde no había tritones (por tanto tiene una especificidad de 110/128=0.859).

Si elegimos como *cutoff* el valor de $C = 0.6$:

```
C <- 0.4
gcnp %>%
  mutate(clasif=ifelse(prob<=C,0,1)) %>%
  tabyl(presence,clasif) %>%
  adorn_totals("col") %>%
  adorn_title(col_name="Clasificación")
```

```
          Clasificación
presence              0  1 Total
       0             95 33   128
       1             25 47    72
```

Con esta nueva regla de decisión la sensibilidad es 47/72=0.653, y la especificidad es 0.74; por tanto hemos conseguido aumentar la sensibilidad a costa de reducir la especificidad. Podríamos construir una tabla con los valores de sensibilidad y especificidad que se consiguen para distintos *cutoffs*:

```
tb <- NULL
for (C in seq(0.1,0.9,by=0.1)){
  espSen <- gcnp %>%
  mutate(clasif=ifelse(prob<=C,0,1)) %>%
  tabyl(presence,clasif) %>%
  adorn_totals("col")
  especificidad <-  espSen[1,2]/espSen[1,4]
  sensibilidad <- espSen[2,3]/espSen[2,4]
  tb <- rbind(tb,c(C,sensibilidad,especificidad))
}
tb <- data.frame(tb)
names(tb) <- c("Cutoff","Sensibilidad","Especificidad")
tb %>% knitr::kable() %>% kableExtra::kable_classic(full_width=FALSE)
```

| Cutoff | Sensibilidad | Especificidad |
|---|---|---|
| 0.1 | 0.9861111 | 0.1562500 |
| 0.2 | 0.8750000 | 0.3984375 |
| 0.3 | 0.7916667 | 0.5937500 |
| 0.4 | 0.6527778 | 0.7421875 |
| 0.5 | 0.4861111 | 0.8593750 |
| 0.6 | 0.3194444 | 0.8984375 |
| 0.7 | 0.1805556 | 0.9765625 |
| 0.8 | 0.0555556 | 0.9921875 |
| 0.9 | 0.0138889 | 1.0000000 |

Como vemos, no hay ningún cutoff para el cual se consigan *simultáneamente* altos valores de sensibilidad y especificidad. Si representamos los valores de sensibilidad frente a la especificidad de la tabla anterior se obtiene la llamada *curva ROC* (receiver operating curve).

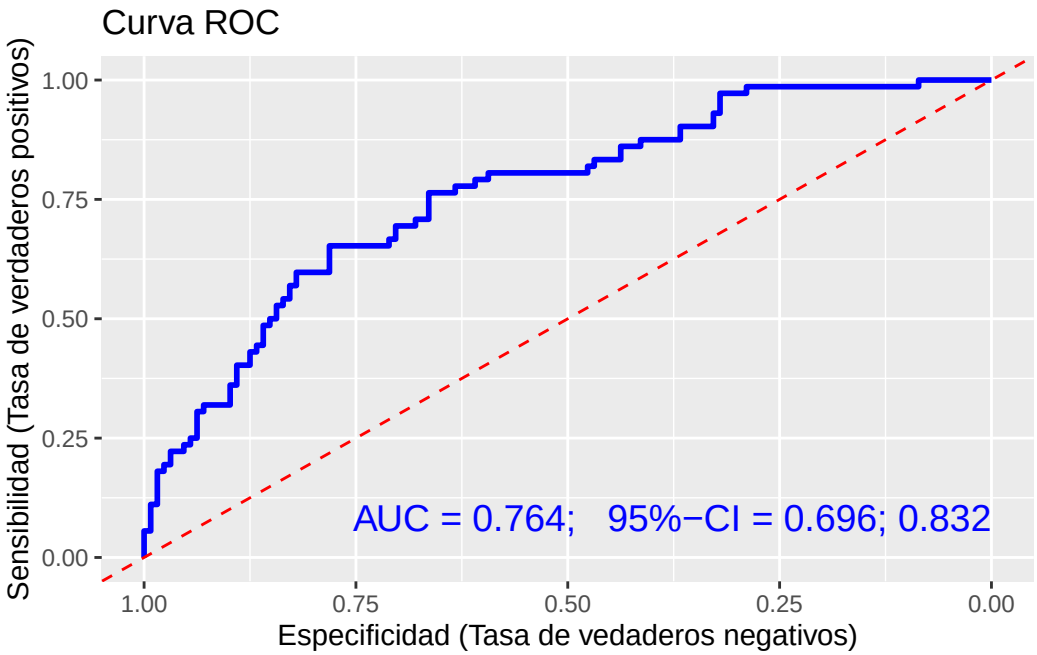
Es usual en la curva ROC representar los valores del eje X (especificidad) en orden inverso (el 1 a la izquierda y el 0 a la derecha). El paquete `pROC` de R dibuja la curva ROC mediante la siguiente sintaxis:

```
library(pROC)
probabilidades_predichas <- predict(mdl1.both,type="response")

# Calculamos la curva ROC:
roc_obj <- roc(gcn$presence, probabilidades_predichas)

# Área bajo la curva
ci <- as.numeric(ci.auc(roc_obj)) # Valores numéricos del área bajo la curva ROC
                                  # y su intervalo de confianza
AUC.ci <- sprintf("AUC = %.3f;   95%%-CI = %.3f; %.3f",ci[2],ci[1],ci[3])

# Gráfica de la curva ROC
ggroc(roc_obj, color = "blue", size = 1) +
  labs(title = "Curva ROC", x = "Especificidad (Tasa de vedaderos negativos)",
       y = "Sensibilidad (Tasa de verdaderos positivos)") +
  geom_abline(slope = 1, intercept = 1, linetype = "dashed", color = "red") +
  annotate("text",x=min(roc_obj$sensitivities),y=min(roc_obj$specificities),
           label=AUC.ci,hjust=1,vjust=-1,size=5,color="blue")
```




El valor AUC significa *Area Under the Curve* y representa justamente el área bajo la curva ROC. Téngase en cuenta que una excelente regla de clasificación sería aquella cuyas sensibilidad y especificidad se aproximaran simultáneamente a 1. Si el modelo logístico ajustado fuese tal que existiera un cutoff para el cual sensibilidad y especificidad se aproximaran ambas a 1, la curva ROC se aproximaría al punto (1,1) situado arriba a la izquierda, y por tanto el área bajo la curva se aproximaría a 1. Por el contrario si la curva ROC se queda "pegada" o muy cerca de la diagonal, ello significa que no hay ningún *cutoff* que permita hacer buenas predicciones; el área bajo la curva ROC en este caso es 0.5. Por ello el área bajo la curva ROC se utiliza como indicador de la capacidad predictiva de la regresión logística.

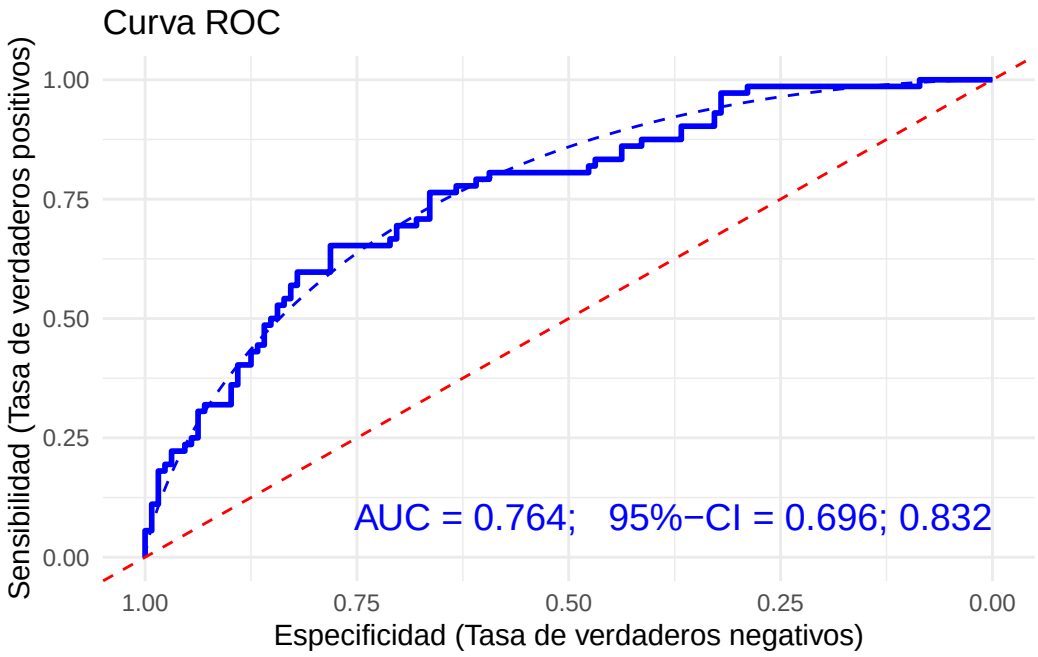
En general, un AUC de 0.5 a 0.6 sugiere que el modelo no tiene capacidad discriminante (es decir, las variables no contienen información para decidir si en la charca habrá tritones o no), un AUC de 0.6 a 0.7 indica una baja capacidad de discriminación, un AUC de 0.7 a 0.8 se considera aceptable, de 0.8 a 0.9 se considera excelente, y más de 0.9 se considera sobresaliente.

En nuestro ejemplo concreto hemos obtenido un AUC de 0.764, por lo que podemos considerar que nuestro modelo tiene una capacidad de clasificación aceptable.

En ocasiones resulta de interés añadir la curva ROC suavizada:

```
# Curva ROC suavizada:
sm <- smooth(roc_obj)
smroc <- data.frame(sensibilidad=sm$sensitivities,especificidad=sm$specificities)

# Gráfica de la curva ROC
ggroc(roc_obj, color = "blue", size = 1) +
  labs(title = "Curva ROC", x = "Especificidad (Tasa de verdaderos negativos)",
       y = "Sensibilidad (Tasa de verdaderos positivos)") +
  theme_minimal() +
  geom_line(data=smroc,aes(x=especificidad,y=sensibilidad),color="blue",lty=2)+
  scale_x_reverse()+
  geom_abline(slope = 1, intercept = 1, linetype = "dashed", color = "red") +
  annotate("text",x=min(roc_obj$sensitivities),y=min(roc_obj$specificities),
           label=AUC.ci,hjust=1,vjust=-1,size=5,color="blue")
```




## 9. Ajsute del modelo usando solo la variable HSI

Si solo usamos como variable predictora la variable HSI:

```
mdl2 <- glm(presence~HSI, data=gcn, family=binomial)
summary(mdl2)
```

```
Call:
glm(formula = presence ~ HSI, family = binomial, data = gcn)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -7.257      1.376  -5.274 1.34e-07 ***
HSI            9.849      1.979   4.977 6.47e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 261.37  on 199  degrees of freedom
Residual deviance: 228.67  on 198  degrees of freedom
AIC: 232.67

Number of Fisher Scoring iterations: 4
```

El valor de la pseudo-$R^2$ es:

```
1-mdl2$deviance/mdl2$null.deviance
```

```
[1] 0.1251143
```

Por tanto este modelo ajusta algo peor que el anterior, que tenía un valor de pseudo-$R^2$ de 0.164. Asimismo el valor de AIC de este modelo es 232.67, mientras que en el modelo anterior era 230.52, lo cual indica que el modelo anterior es ligeramente mejor, aunque al ser la diferencia en AIC menor de 3, ambos modelos pueden considerarse prácticamente equivalentes. En lo que se refiere al área bajo la curva ROC, para este modelo obtenemos:

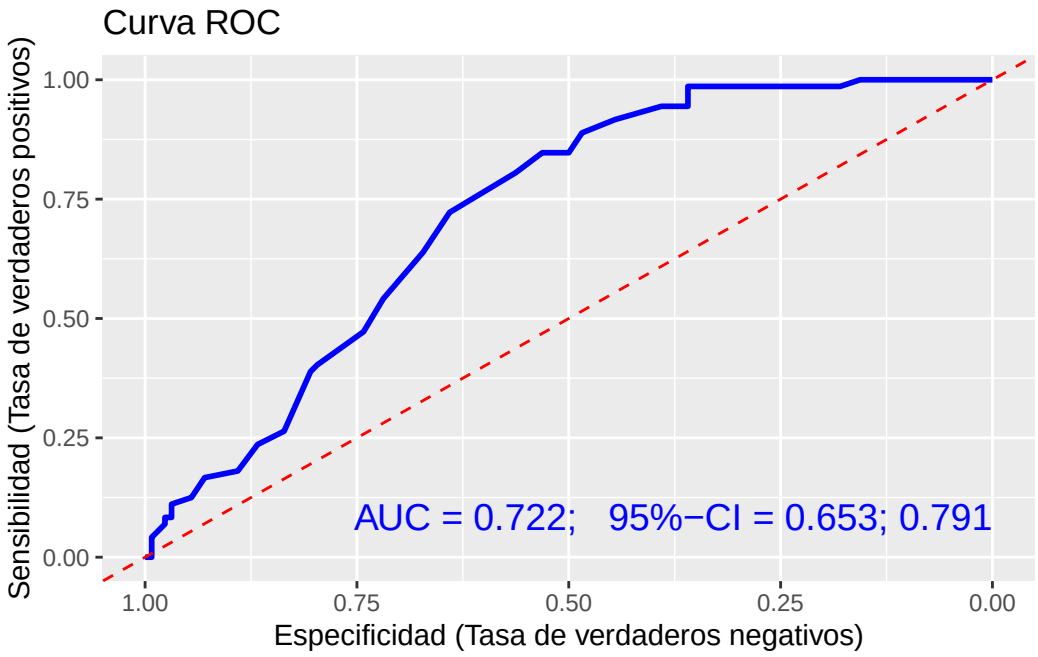
```
probabilidades_predichas <- predict(mdl2,type="response")

# Calculamos la curva ROC:
```

```
roc_obj2 <- roc(gcn$presence, probabilidades_predichas)

# Área bajo la curva
ci <- as.numeric(ci.auc(roc_obj2)) # Valores numéricos del área bajo la curva ROC
                                   # y su intervalo de confianza
AUC.ci <- sprintf("AUC = %.3f;   95%%-CI = %.3f; %.3f",ci[2],ci[1],ci[3])

# Gráfica de la curva ROC
ggroc(roc_obj2, color = "blue", size = 1) +
  labs(title = "Curva ROC", x = "Especificidad (Tasa de verdaderos negativos)",
       y = "Sensibilidad (Tasa de verdaderos positivos)") +
  geom_abline(slope = 1, intercept = 1, linetype = "dashed", color = "red") +
  annotate("text",x=min(roc_obj$sensitivities),y=min(roc_obj$specificities),
           label=AUC.ci,hjust=1,vjust=-1,size=5,color="blue")
```




El área bajo la curva es también muy parecida a la obtenida con el modelo anterior, solapándose en gran medida sus intervalos de confianza.

En definitiva, podemos concluir que el modelo que solo tiene el HSI como variable explicativa es prácticamente equivalente al modelo anterior en el que hemos usado cinco variables. Usar un modelo u otro dependerá en definitiva del interés último del investigador. Para calcular el HSI es necesario medir 9 variables y combinar sus valores;

por tanto desde el punto de vista del esfuerzo de toma de datos para la evaluación, resultará algo más costoso un modelo basado en HSI que un modelo basado en las cinco variables que hemos considerado. Sin embargo, desde un punto de vista de simplificación y comunicación de resultados, puede resultar más sencillo utilizar un modelo que emplee solo el HSI, pues con un único valor se cualifica la capacidad del ecosistema para soportar una población de tritones.