Modelización de la presencia de *Fasciola hepatica* en granjas de vacuno gallegas

Gabriel Carbonell, Zaira García, Celia Sifre

16/6/2022

Introducción

La *Fasciola hepatica* es un gusano parásito con ciclo biológico de dos generaciones en dos hospedadores distintos: un molusco (como los caracoles) y un mamífero. Su presencia o ausencia es dependiente de las condiciones geográficas y climatológicas, además del estado de salud del animal. Para su prevención existen métodos faracológicos, químicos y de alteración del entorno.

Se puede tratar de modelizar la presencia o ausencia de dicho parásito a partir de la información obtenida sobre 400 granjas de vacuno gallegas. Concretamente, se cuenta con diez variables relativas a información geográfica y climatológica de dichas granjas, además de su densidad y el tipo de vaca.

Descripción de los datos

Realizaremos un análisis descriptivo del conjunto de datos que se empleará. Como se ha mencionado, contiene 400 observaciones relativas a 10 variables, las cuales se especifican a continuación.

Variable respuesta:

- `InfFasc`: presencia o ausencia del parásito *Fasciola hepatica* (0 = no, 1 = sí; binaria).

Variables explicativas:

- `Age`: edad de la vaca, en años (cuantitativa discreta).
- `Aptitude`: un factor de dos niveles que define si la vaca se dedica a la producción de leche (L) o de carne (C) (categórica nominal).
- `Rainfall`: pluviometría de la granja (cuantitativa continua).
- `Altitude`: altitud a la que se encuentra la granja (cuantitativa continua).
- `Slope`: pendiente en el punto en que se encuentra la granja (cuantitativa continua).
- `Density`: densidad de la granja (cuantitativa continua).
- `X` e `Y`: se trata de las coordenadas geográficas de la granja (cuantitativa continua).

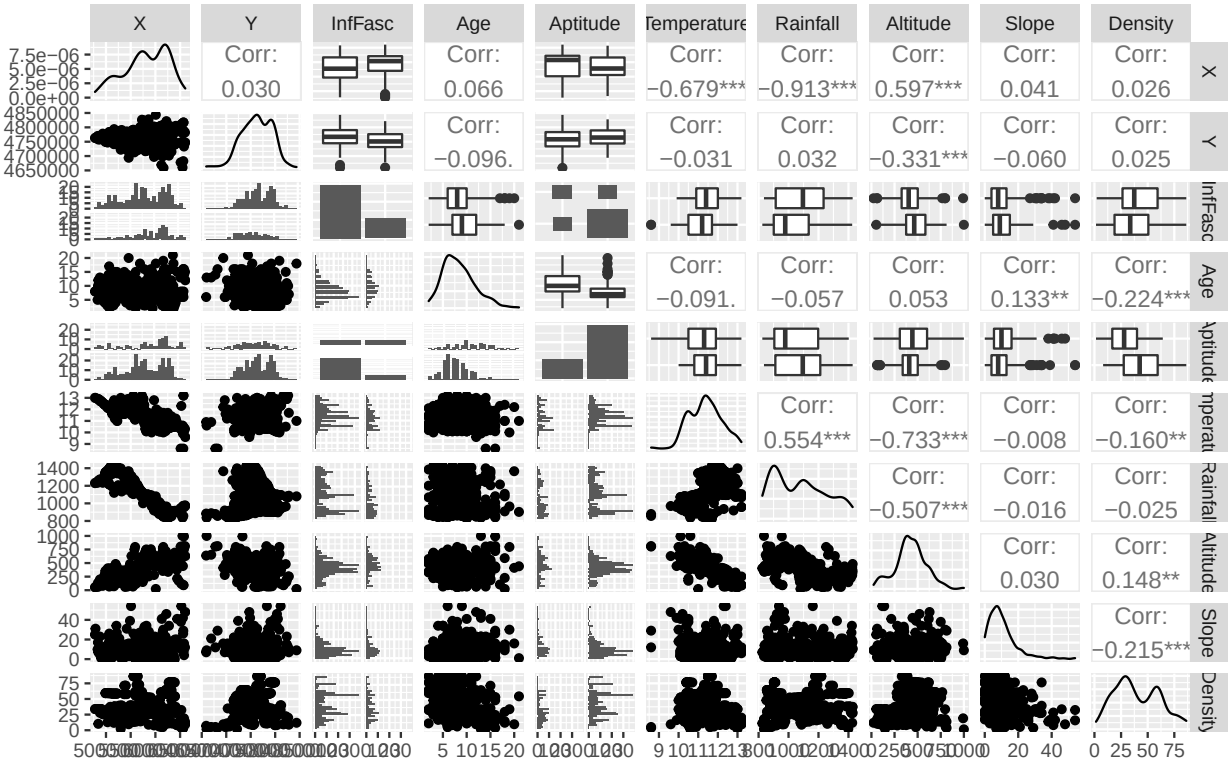
Empezaremos con un análisis descriptivo numérico:

```
##        X               Y               Age          Temperature
##  Min.   :477683   Min.   :4659039   Min.   : 2.000   Min.   : 8.60
##  1st Qu.:555098   1st Qu.:4741653   1st Qu.: 6.000   1st Qu.:10.70
##  Median :581830   Median :4763138   Median : 8.000   Median :11.40
##  Mean   :581354   Mean   :4762492   Mean   : 8.375   Mean   :11.38
##  3rd Qu.:617827   3rd Qu.:4787682   3rd Qu.:10.000   3rd Qu.:11.90
##  Max.   :660537   Max.   :4843638   Max.   :21.000   Max.   :13.20
```

```
##     Rainfall         Altitude          Slope          Density
##  Min.   : 825.0   Min.   :  23.0   Min.   : 0.00   Min.   : 1.468
##  1st Qu.: 909.8   1st Qu.: 336.5   1st Qu.: 4.75   1st Qu.:25.641
##  Median :1075.0   Median : 422.5   Median : 8.00   Median :35.282
##  Mean   :1078.1   Mean   : 425.1   Mean   :10.43   Mean   :40.407
##  3rd Qu.:1206.2   3rd Qu.: 521.5   3rd Qu.:13.25   3rd Qu.:58.667
##  Max.   :1428.0   Max.   :1000.0   Max.   :54.00   Max.   :86.276
```

```
##  InfFasc     Aptitude
##  0:292      C    :107
##  1:108      L    :293
```

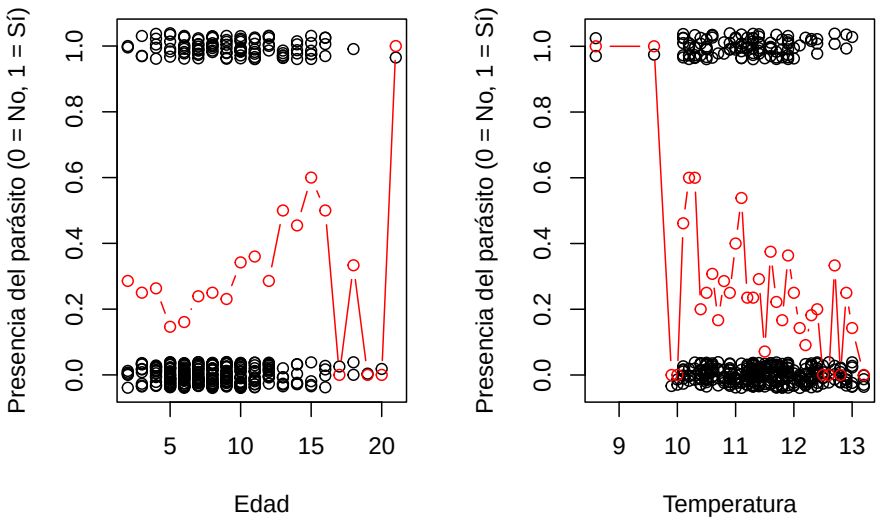
También podemos realizar un análisis descriptivo gráfico. En primer lugar, generamos un gráfico de tipo *pairs*, incluyendo la correlación entre variables.



Se observa como las correlaciones más altas se dan entre:

- la longitud (X) con la temperatura (< -0.6), la pluviometría (< -0.9) y la altitud (> 0.5).
- La temperatura con la pluviometría y la altitud.
- La pluviometría con la altitud.

Por otra parte, si nos fijamos en los diagramas de cajas entre la variable respuesta y la explicativa, vemos que la media suele ser parecida excepto en el caso de la pluviometría, lo cual podría deberse al hecho de que la presencia de agua es imprescindible en el ciclo vital del parásito.



Por último, si se valora la relación media de la variable respuesta con respecto a las variables respuesta, las variables con un compotamiento más destacable son la edad de las vacas y la temperatura. Como vemos, a medida que aumenta la edad, la probabilidad de la presencia del parásito crece, pero a los 16 baja y luego tiene una bajada abrupta. Una razón posible sería que la presencia del parásito reduce la esperanza de vida de las vacas. En el caso de temperatura, a medida que esta aumenta decrece la presencia media del parásito. Esto se debe a que los aumentos de tempeatura correlacionan con el aumento de temperatura del agua y su evaporación, reduciendo la viabilidad del parásito en su estado larvario. Se pueden encontrar los gráficos de este tipo con respecto a todas las variables cuantitativas en el material suplementario.

Modelización

Una vez realizado este análisis previo daremos paso a la modelización de dicho problema.

Nuestra variable respuesta InfFasc_i indica la presencia (1) o ausencia (0) del parásito en las $i = 1, \ldots, 400$ vacas analizadas.

Como InfFasc es una variable binaria se distribuye como una Bernoulli de parámetro π_i, donde π_i es la probabilidad de que una vaca i tenga el parásito:

$$\text{InfFasc}_i \sim \text{Ber}(\pi_i), \quad i = 1, \ldots, 400.$$

A la hora de elegir la mejor transformación para relacionar el predictor lineal con la respuesta media, compararemos los resultados sobre las transformaciones logit, probit y cloglog.

```
ajuste.logit <- glm(InfFasc ~ ., data=datos, family = binomial(link="logit"))
ajuste.probit <- glm(InfFasc ~ ., data=datos, family = binomial(link="probit"))
ajuste.cloglog <- glm(InfFasc ~ ., data=datos, family = binomial(link="cloglog"))
```

Si comparamos la Deviance y el AIC asociados a cada una de las tres alternativas, obtenemos en ambos casos que el modelo que mejor resultado nos ofrece es la transformación logit.

```
##                      AIC Deviance
## ajuste.logit   408.3946 388.3946
## ajuste.probit  408.4712 388.4712
## ajuste.cloglog 413.6585 393.6585
```

Por tanto, a partir de ahora trabajaremos con la transformación logit para buscar la mejor selección de variables explicativas y también para valorar la capacidad predictiva de nuestro modelo.

Cuando trabajamos con datos binarios, la opción más adecuada para valorar si el ajuste que hemos realizado es correcto es el test de Hosmer y Lemeshow. Al realizar dicho test obtenemos un p-valor de 0.401, por tanto, asumimos que el ajuste es bueno. Como hemos hecho el test para el modelo completo y se cumplen las condiciones de aplicabilidad entonces para cualquier submodelo que estimemos también se cumplirán.

Para seleccionar las variables de nuestro modelo hemos realizado un stepwise y hemos obtenido que las variables significativas son: Y, Aptitude, Temperature y Altitude. Así, la relación entre la respuesta media y el predictor lineal tiene la siguiente expresión:

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 \text{Y}_i + \beta_2 \text{Aptitude}_i + \beta_3 \text{Temperature}_i + \beta_4 \text{Altitude}_i$$

Al estimar este modelo obtenemos un AIC más bajo de 403.06 y una Deviance un pelín más alta de 393.06. Aunque en este estudio trabajaremos con la inferencia frecuentista, queremos plantear un modelo con algunas de las variables de nuestra base de datos (un factor y una covariable) con el lenguaje de programación WinBUGS.

```
modelo1 <- function(){
  for(i in 1:N){
  InfFasc[i] ~ dbern(p[i])
    logit(p[i]) <- beta1 + beta2*Temperature[i] + beta.A[Aptitude[i]]
  }
  # distribuciones previas
  beta1 ~ dflat()
  beta2 ~ dflat()
  beta.A[2] ~ dflat()
  #restricción
  beta.A[1] <- 0
}
set.seed(1)
datos1 <- list(InfFasc=datos$InfFasc, Temperature=datos$Temperature,
               Aptitude=as.numeric(datos$Aptitude),
               N=dim(datos)[1])
iniciales1 <- function(){
  list(beta1=rnorm(1,0,3), beta2=rnorm(1), beta.A=c(NA,rnorm(1)))}
parametros1 <- c("beta1", "beta2", "beta.A")
ResulModelo1 <- bugs(model = modelo1, data = datos1, inits = iniciales1,
                     param = parametros1, n.iter = 20000, n.burnin = 2000)
```

Al ejecutar este modelo y realizar inferencia bayesiana, obtenemos resultados muy equiparables con el que realizado con la inferencia frecuentista. Se pueden consultar en el material suplementario, donde además ejecutamos un modelo GLM con las mismas variables, permitiendo comprobar que los resultados son equiparables.

Ahora partiremos del modelo obtenido mediante *step* y estudiaremos relaciones no lineales entre la variable respuesta y las covariables para intentar mejorarlo tanto en términos de ajuste como en términos de como predicen.

En primer lugar, hemos añadido las variables con suavizado que no eran significativas cuando hemos realizado el stepwise. En todas las covariables hemos obtenido que los grados de libertad eran prácticamente 1 y además al realizar el gráfico del intervalo de confianza este contiene el valor 0 casi por todas partes. Por este motivo, decidimos no incluir estas covariables en nuestro modelo.

Entonces, tenemos como covariables las mismas que nos recomendaba el método stepwise: `Y`, `Aptitude`, `Temperature` y `Altitude`. Pero claro, en nuestras variables tenemos las coordenadas de localización `X` e `Y`. Es por ello que nos hemos planteado un nuevo modelo con estas covariables y un suavizado bivariante de `X` e `Y`. Al realizar este modelo, obtenemos un R^2 ajustado de 0.286 y una deviance explicada del 30.2%.

Para mejorar este último modelo, pensamos que al tener como covariable la altitud de la granja también podemos pensar en suavizarla. Cuando hemos estimado este modelo, hemos obtenido un R^2 ajustado un ligeramente más elevado de 0.3 y una *deviance* explicada del 31.5%.

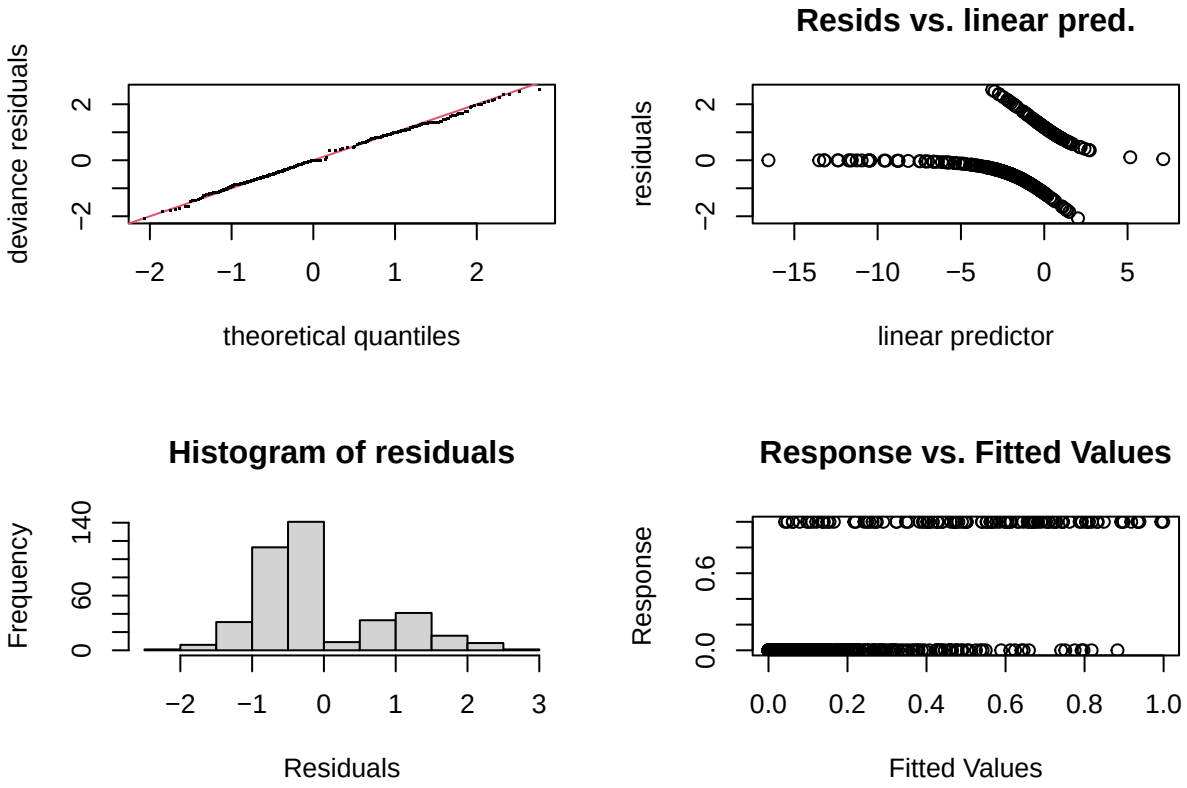
Finalmente el mejor modelo que hemos encontrado es el siguiente:

```
ajuste.gam1 <- gam(InfFasc ~ s(X,Y) + Aptitude + te(Altitude, Rainfall), data=datos,
                   family = binomial(link="logit"))
summary(ajuste.gam1)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## InfFasc ~ s(X, Y) + Aptitude + te(Altitude, Rainfall)
##
## Parametric coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -0.6646     0.3634  -1.829   0.0674 .
## Aptitude   L      -1.7012     0.3493  -4.870 1.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf Ref.df Chi.sq p-value
## s(X,Y)                 27.034 28.515  42.33 0.04860 *
## te(Altitude,Rainfall)  5.541  6.857  18.95 0.00721 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.313   Deviance explained = 32.9%
## UBRE = -0.044944  Scale est. = 1         n = 400
```

Donde el R^2 ajustado vale 0.313 y el porcentaje de la deviance explicada es del 32.9%. Además, vemos que los grados de libertad para las dos variables suavizadas son menores que k-1 (en este caso utilizamos el valor de k por defecto), así que realmente el suavizado si que mejora el modelo. Antes de concluir comprobaremos que cumple las condiciones de aplicabilidad y después analizaremos la calidad de predicción.

A parte, también nos planteamos añadir el suavizado bivariante de las variables temperatura y altitud, pero finalmente no mejoraba el modelo ni en términos de ajuste ni en calidad de predicción.

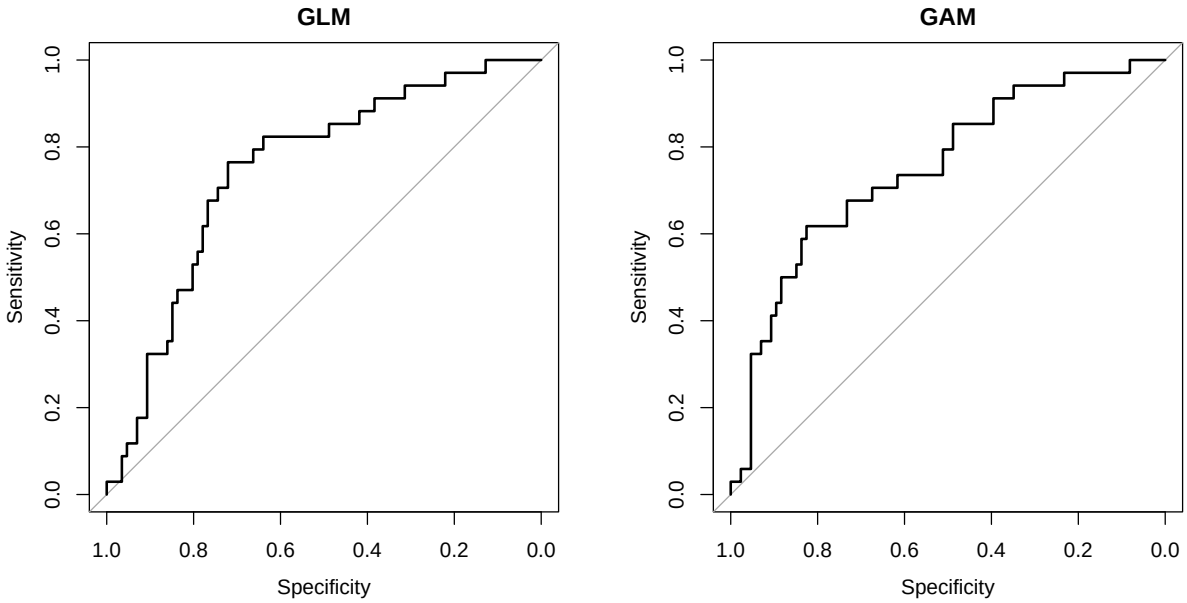



```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-1.051191e-06,1.992387e-07]
## (score -0.04494449 & scale 1).
## Hessian positive definite, eigenvalue range [0.000112105,0.003242832].
## Model rank =  55 / 55
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                          k'   edf k-index p-value
## s(X,Y)                29.00 27.03    1.04    0.88
## te(Altitude,Rainfall) 24.00  5.54    1.00    0.62
```

Observamos que se cumple la normalidad (gráfica superior izquierda), además los residuos deviance se encuentran entre -2 y 2 (gráfica superior derecha). Lo único que nos podría preocupar es que no tenemos una simetría clara en el histograma de los residuos, pero vemos que prácticamente están centrados en 0. Así, concluimos que el modelo cumple las condiciones de aplicabilidad. Además, observamos que el *k-index* es mayor que 1 y el p-valor mayor que 0.05, así los valores de k que nos da la función por defecto son adecuados para el suavizado de estas variables.

Evaluación de la capacidad predictiva de los modelos frecuentistas

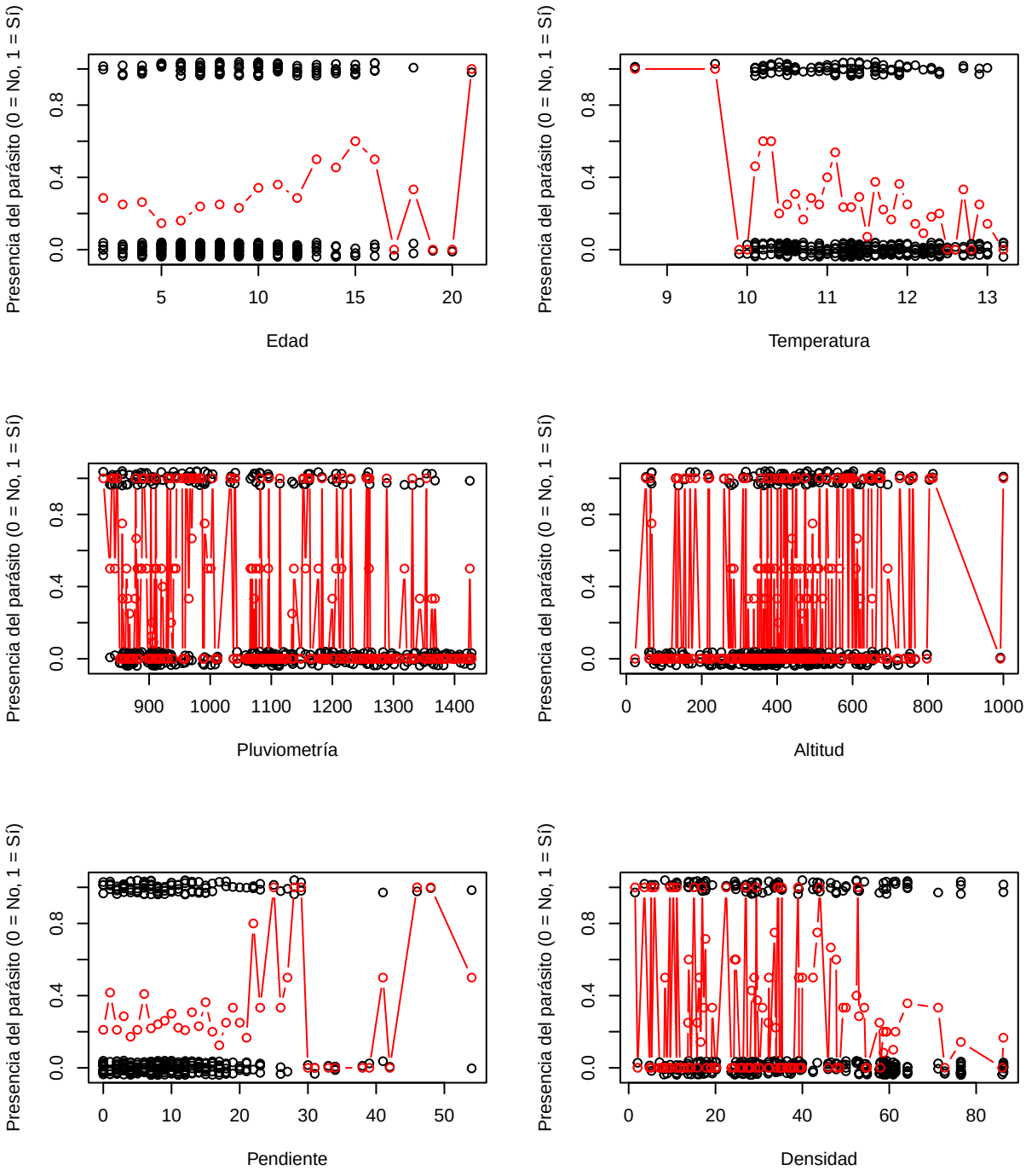
A continuación, realizaremos una valoración de la capacidad predictiva del modelo obtenido con *step* y del modelo aditivo mediante la división de los datos en una parte de entrenamiento (70%) y otra de testeo (30%). En primer lugar hemos obtenido la curva ROC de los modelos, obteniendo un área bajo la curva de 0.75 con un intervalo de confianza del 95% de 0.67 a 0.83 para el modelo aditivo.



Además, decidimos obtener más estadísticos para decidirnos por un modelo u otro. Respecto al resto de estadísticos, la sensibilidad de ambos modelos es igual (90%), pero el modelo aditivo supera al GLM con una exactitud del 75%, especificidad de 41%, valor predictivo positivo del 79% y valor predictivo negativo de 0.61. En el caso de la especificidad, esta es muy superior en el aditivo (la del GLM es 17%). Por ello, a nivel predictivo nos inclinamos por elegir el modelo aditivo sobre el GLM.

Material suplementario

Figuras



Resultados de WinBugs y comparación con un modelo frecuentista equivalente

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
beta1	5.556	1.682	2.511	4.373	5.527	6.629	9.033	1.059	1000
beta2	-0.484	0.148	-0.792	-0.577	-0.481	-0.380	-0.213	1.005	1000
beta.A[2]	-1.619	0.238	-2.101	-1.766	-1.632	-1.449	-1.186	1.001	1000
deviance	414.664	2.395	412.100	413.000	414.100	415.575	421.897	1.008	320

```
##
## Call:
## glm(formula = InfFasc ~ Aptitude + Temperature, family = binomial(link = "logit"),
##     data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5068  -0.7170  -0.5785   0.9201   2.1778
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.4559     1.7298   3.154  0.00161 **
## Aptitude   L      -1.5944     0.2514  -6.342 2.26e-10 ***
## Temperature       -0.4756     0.1523  -3.123  0.00179 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 466.61  on 399  degrees of freedom
## Residual deviance: 411.80  on 397  degrees of freedom
## AIC: 417.8
##
## Number of Fisher Scoring iterations: 4
```

Calidad de precicción del modelo GLM

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 78 28
##          1  8  6
##
##                Accuracy : 0.7
##                  95% CI : (0.6096, 0.7802)
##     No Information Rate : 0.7167
##     P-Value [Acc > NIR] : 0.697456
##
##                   Kappa : 0.1015
##
##  Mcnemar's Test P-Value : 0.001542
##
##             Sensitivity : 0.9070
##             Specificity : 0.1765
##          Pos Pred Value : 0.7358
##          Neg Pred Value : 0.4286
##              Prevalence : 0.7167
##          Detection Rate : 0.6500
##    Detection Prevalence : 0.8833
##       Balanced Accuracy : 0.5417
##
##        'Positive' Class : 0
##
```

Calidad de precicción del modelo GAM

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 77 20
##          1  9 14
##
##                Accuracy : 0.7583
##                  95% CI : (0.6717, 0.8318)
##     No Information Rate : 0.7167
##     P-Value [Acc > NIR] : 0.18162
##
##                   Kappa : 0.3404
##
##  Mcnemar's Test P-Value : 0.06332
##
##             Sensitivity : 0.8953
##             Specificity : 0.4118
##          Pos Pred Value : 0.7938
##          Neg Pred Value : 0.6087
##              Prevalence : 0.7167
```

```
##          Detection Rate : 0.6417
##    Detection Prevalence : 0.8083
##       Balanced Accuracy : 0.6536
##
##        'Positive' Class : 0
##
```

Código

```
library(GGally)
library(dplyr)
library(tidyr)
library(mgcv)
library(psych)
library(knitr)
library(R2WinBUGS)
library(caret)
library(pROC)
library(ResourceSelection)

# Carga de los datos
datos <- read.csv("data_galicia.txt", sep = " ")
datos$Aptitude <- as.factor(datos$Aptitude)
datos$InfFasc <- as.factor(datos$InfFasc)

attach(datos)
summary(datos[, c(1, 2, 4, 6, 7, 8, 9, 10)])
summary(datos[, -c(1, 2, 4, 6, 7, 8, 9, 10)])


ggpairs(datos)

par(mfrow=c(3,2))
plot(as.numeric(Age), jitter(as.numeric(InfFasc)-1, factor = 0.2),
     xlab = "Edad",
     ylab = "Presencia del parásito (0 = No, 1 = Sí)")
tt <- table(InfFasc, Age)
prc <- tt[2,] / (tt[1,] + tt[2,])
lines(as.numeric(colnames(tt)), prc,type='b', col='RED')

plot(as.numeric(Temperature), jitter(as.numeric(InfFasc)-1, factor = 0.2),
     xlab = "Temperatura",
     ylab = "Presencia del parásito (0 = No, 1 = Sí)")
tt <- table(InfFasc, Temperature)
prc <- tt[2,] / (tt[1,] + tt[2,])
lines(as.numeric(colnames(tt)), prc,type='b', col='RED')

plot(as.numeric(Rainfall), jitter(as.numeric(InfFasc)-1, factor = 0.2),
     xlab = "Pluviometría",
     ylab = "Presencia del parásito (0 = No, 1 = Sí)")
tt <- table(InfFasc, Rainfall)
prc <- tt[2,] / (tt[1,] + tt[2,])
lines(as.numeric(colnames(tt)), prc,type='b', col='RED')

plot(as.numeric(Altitude), jitter(as.numeric(InfFasc)-1, factor = 0.2),
     xlab = "Altitud",
     ylab = "Presencia del parásito (0 = No, 1 = Sí)")
tt <- table(InfFasc, Altitude)
prc <- tt[2,] / (tt[1,] + tt[2,])
lines(as.numeric(colnames(tt)), prc,type='b', col='RED')
```

```
plot(as.numeric(Slope), jitter(as.numeric(InfFasc)-1, factor = 0.2),
     xlab = "Pendiente",
     ylab = "Presencia del parásito (0 = No, 1 = Sí)")
tt <- table(InfFasc, Slope)
prc <- tt[2,] / (tt[1,] + tt[2,])
lines(as.numeric(colnames(tt)), prc,type='b', col='RED')

plot(as.numeric(Density), jitter(as.numeric(InfFasc)-1, factor = 0.2),
     xlab = "Densidad",
     ylab = "Presencia del parásito (0 = No, 1 = Sí)")
tt <- table(InfFasc, Density)
prc <- tt[2,] / (tt[1,] + tt[2,])
lines(as.numeric(colnames(tt)), prc,type='b', col='RED')

ajuste.logit <- glm(InfFasc ~ ., data=datos, family = binomial(link="logit"))
ajuste.probit <- glm(InfFasc ~ ., data=datos, family = binomial(link="probit"))
ajuste.cloglog <- glm(InfFasc ~ ., data=datos, family = binomial(link="cloglog"))

x <- matrix(c(AIC(ajuste.logit), deviance(ajuste.logit),
              AIC(ajuste.probit), deviance(ajuste.probit),
              AIC(ajuste.cloglog), deviance(ajuste.cloglog)), nrow=3,byrow=T)
colnames(x) <- c("AIC","Deviance")
rownames(x) <- c("ajuste.logit", "ajuste.probit", "ajuste.cloglog")
x

hoslem.test(as.numeric(as.character(datos$InfFasc)), fitted(ajuste.logit))

maximo <- glm(InfFasc ~ ., family = binomial("logit"), data = datos)
minimo <- glm(InfFasc ~ 1, family = binomial("logit"), data = datos)
ajuste.step1 <- step(maximo, direction = "backward", scope = minimo)
ajuste.step1$coefficients

ajuste.step <- glm(InfFasc ~ Y + Aptitude + Temperature + Altitude, data=datos,
            family = binomial(link="logit"))
summary(ajuste.step)


modelo1 <- function(){
  for(i in 1:N){
  InfFasc[i] ~ dbern(p[i])
    logit(p[i]) <- beta1 + beta2*Temperature[i] + beta.A[Aptitude[i]]
  }
  # distribuciones previas
  beta1 ~ dflat()
  beta2 ~ dflat()
  beta.A[2] ~ dflat()
  #restricción
  beta.A[1] <- 0
}

set.seed(1)
datos1 <- list(InfFasc=datos$InfFasc, Temperature=datos$Temperature,
               Aptitude=as.numeric(datos$Aptitude),
```

```
              N=dim(datos)[1])
iniciales1 <- function(){
  list(beta1=rnorm(1,0,3), beta2=rnorm(1), beta.A=c(NA,rnorm(1)))
  }
parametros1 <- c("beta1", "beta2", "beta.A")
ResulModelo1 <- bugs(model = modelo1, data = datos1, inits = iniciales1,
                     param = parametros1, n.iter = 20000, n.burnin = 2000)

ajuste.gam1 <- gam(InfFasc ~ s(X,Y) + Aptitude + te(Altitude, Rainfall), data=datos,
                   family = binomial(link="logit"))
summary(ajuste.gam1)

par(mfrow = c(2,2))
gam.check(ajuste.gam1)

modelo <- glm(InfFasc ~ Aptitude + Temperature,
              data = datos,
              family=binomial(link="logit"))
summary(modelo)

# Análisis de la capacidad predictiva

# Dividimos los datos
set.seed(123)
split <- sample(c(rep(0, 0.7 * nrow(datos)), rep(1, 0.3 * nrow(datos))))
train <- datos[split == 0, ]
test <- datos[split == 1, ]

# Estadísticos de predicción para el GLM
train.glm <- glm(InfFasc ~ Y + Aptitude + Temperature + Altitude, data=train,
                 family = binomial(link="logit"))

p.glm <- predict(train.glm, newdata=test, type='response')
result.glm <- as.factor(ifelse(p.glm > 0.5, 1, 0))
confusionMatrix(data = result.glm, reference = test$InfFasc)
curv_roc.glm <- roc(as.factor(test$InfFasc), p.glm)

# Estadísticos de predicción para el GAM
train.gam <- gam(InfFasc ~ s(X,Y) + Aptitude + te(Altitude, Rainfall), data=train,
                 family = binomial(link="logit"))

p.gam <- predict(train.gam, newdata=test, type='response')
result.gam <- as.factor(ifelse(p.gam > 0.5, 1, 0))
confusionMatrix(data=result.gam, reference = test$InfFasc)
curv_roc.gam <- roc(as.factor(test$InfFasc), p.gam)

# Representación curvas ROC
par(mfrow=c(1, 2))
plot(curv_roc.glm, xlim=c(1,0), main = "GLM")
plot(curv_roc.gam, xlim=c(1,0), main = "GAM")
```