# Modelling multi-stressor responses across ecosystems in R

*Cayetano Gutiérrez Cánovas & Pol Capdevila Lanzaco*

*2019-01-29*

## Resumen

Este taller pretende iniciar a los participantes en el modelado de los efectos de estresores múltiples en los ecosistemas. El cambio global está incrementando el número y la intensidad de los estresores que están impactando los sistemas naturales, que pueden producir efectos que se desvían de la simple suma de sus efectos individuales (interacciones). Para cuantificar los efectos combinados de los distintos estresores, es necesario que usemos técnicas que consideren los posibles efectos interactivos y que exploren la importancia relativa de cada estresor.

Durante el taller, los participantes se familiarizarán con el protocolo de modelado para explorar los efectos de los estresores múltiples usando código escrito en R. Este protocolo consiste en dos pasos principales:

1. Un análisis exploratorio (Random Forest y Boosted Regression Trees) para evaluar la importancia de los estresores e identificar posibles interacciones.
2. Estimar el tamaño del efecto y la importancia de los estresores y de sus interacciones usando la técnica de "multi-model inference" (promedio ponderado de los coeficientes de los estresores que aparecen en los mejores modelos).

Los participantes tendrán contacto con el tipo de datos y técnicas estadísticas necesarias para modelar los efectos de los estresores múltiples, así como con las funciones de R y el código necesario para su implementación. La filosofía del taller es puramente práctica: trabajaremos un caso de estudio para mostrar el potencial de este enfoque, así como las preguntas más frecuentes que pueden surgir durante la aplicación de este protocolo. Proporcionaremos artículos y código para que los participantes puedan continuar su aprendizaje después del taller. El conocimiento adquirido durante el taller podrá ser aplicado a cualquier tipo de ecosistema u organismo.

Recomendaciones:

- Conocimiento básico de R y modelos lineales generalizados, GLMs (regresiones, ANOVAs).
- Portátil propio con las últimas versiones de R y RStudio previamente instalados.

## Los modelos ecológicos

En ciencias de la vida, y en particular en ecología, usamos modelos para testar hipótesis, analizar patrones o predecir respuestas en distintos tipos de ecosistemas y unidades de organización biológica. Aunque en muchas ocasiones resulta conveniente simplificar la realidad y centrarnos en una variable respuesta y otra explicativa, en general, solemos explorar una lista cada vez mayor de factores candidatos que puedan explicar nuestra variable de interés.

El hecho de usar múltiples variables predictoras ofrece una serie de ventajas, pero también conlleva una mayor complejidad de análisis y ciertas limitaciones prácticas. Entre las ventajas, podemos destacar una mayor capacidad para cuantificar el tamaño del efecto de cada predictor (*effect size*) y su importancia relativa. Sin embargo, a medida que introducimos nuevos predictores, nuestro modelo se va haciendo más complejo y empezarán a surgir ciertas complicaciones. Entre las más habituales, encontramos el reto de encontrar el "mejor" modelo (ver sección 3 para más detalles) o la ocurrencia de interacciones entre nuestras variables.

En este taller mostraremos un protocolo que usa distintas estrategias para abordar el modelado de los efectos de varios predictores en una variable respuesta usando el marco desarrollado para los multiple stressors (Figura 1; Feld et al., 2016). Aunque este taller no pretende ser un repaso exhaustivo a los distintos métodos

de modelado en ecología y ciencias de la vida, veremos un abanico amplio de métodos que se adaptan a los casos más comunes que podemos encontrar. Para profundizar, recomendamos libros especializados en técnicas de modelado ecológico (Burnham and Anderson, 2002; Crawley, 2014; Zuur et al., 2009).

Los modelos estadísticos son herramientas con distintas capacidades, ventajas e inconvenientes, que tienen un ámbito concreto de aplicación y una serie de reglas que debemos cumplir para asegurarnos de que los resultados que obtenemos son fiables. De manera ideal, querríamos una técnica de modelado fuera:

- **Flexible** respecto a su ámbito de aplicación o a los datos que pudieran incluir
- **Transparente**, de sencilla interpretación y con una alta capacidad para identificar las variables predictoras más importantes y predecir respuestas ecológicas.

Pero como ocurre habitualmente, no lo podemos tener todo. Así, hay modelos (modelos generalizados lineales, incluyendo la familia de los GLM y sus extensiones mixtas) que son ampliamente conocidos, para lo bueno y para lo malo. Se trata de modelos muy transparentes, y con una gran capacidad para testar hipótesis ecológicas, al precio de ser bastante sensibles al incumplimiento de ciertas asunciones previas (e.g. la normalidad y homocedasticidad de los residuos del modelo, y la independencia de las observaciones). Además, también tienen una limitación importante respecto al número de predictores que podemos explorar y que depende del número de observaciones de nuestros datos. De manera general, solo se puede testar un predictor por cada 10 observaciones, regla conocida como "one in ten rule" (Harrell, 2001).

Por otra parte, en el extremo opuesto, los modelos basados en árboles de decisión y algoritmos de *machine learning*, conocidos como *Classification and Regression Trees* (CART), son modelos extremadamente flexibles, con una gran capacidad exploratoria y con pocas asunciones previas a cumplir, por lo que son capaces de superar muchas de las limitaciones presentes en los modelos de tipo GLM/GLMM. Es decir, pueden modelar datos de naturaleza muy diversa y acomodando relaciones tanto lineales como no lineales. Además, no están sujetos a las asunciones de normalidad y homocedasticidad de los residuos del modelo. Además, teóricamente son capaces de explorar un gran número de predictores pese a tener pocas observaciones. A cambio, se trata de modelos muy complejos, poco transparentes y no muy adecuados para testar hipótesis ya que no podemos calcular los tamaños de los efectos de cada predictor ni su *p-valor*.

Más allá de cuestiones técnicas, no debemos de olvidar que estamos haciendo ciencia. Así, antes de comenzar a analizar los datos, debemos reflexionar bien sobre la relación que cada uno de nuestros predictores tiene sobre la variable respuesta. Es recomendable establecer relaciones teóricas que tengan un claro sentido biológico para desarrollar hipótesis o predicciones robustas que puedan ser testadas estadísticamente (ver pp. 1-5 Crawley, 2014). Esto ayudará a descartar predictores que puedan tener una relación espuria (sin sentido ecológico) con la variable respuesta y facilitará la interpretación de los resultados. Por lo tanto, antes de analizar toca leer y pensar: es imprescindible que conozcamos bien la literatura clave y el conocimiento más actual (*state of the art*) de nuestro campo.

## Empezando por lo primero: Cargado de datos

Antes de empezar con el taller debemos asegurarnos que cargamos bien los datos y paquetes necesarios para este workshop.

Para ello primero determinamos el directorio de trabajo mediante la función `setwd`.

```
# Setting working directory

setwd("C:/Users/VUESTRO USUARIO/Dropbox/multistress_SIBECOL/Participants")
```

Una vez asignado el directorio de trabajo, vamos a pasar a cargar las librerías. Seguramente la mayoría de ellas las tendremos que instalar, usando la función `install.packages()`.

```
# Install required packages

if (!requireNamespace("BiocManager", quietly = TRUE))
   install.packages("BiocManager")
```

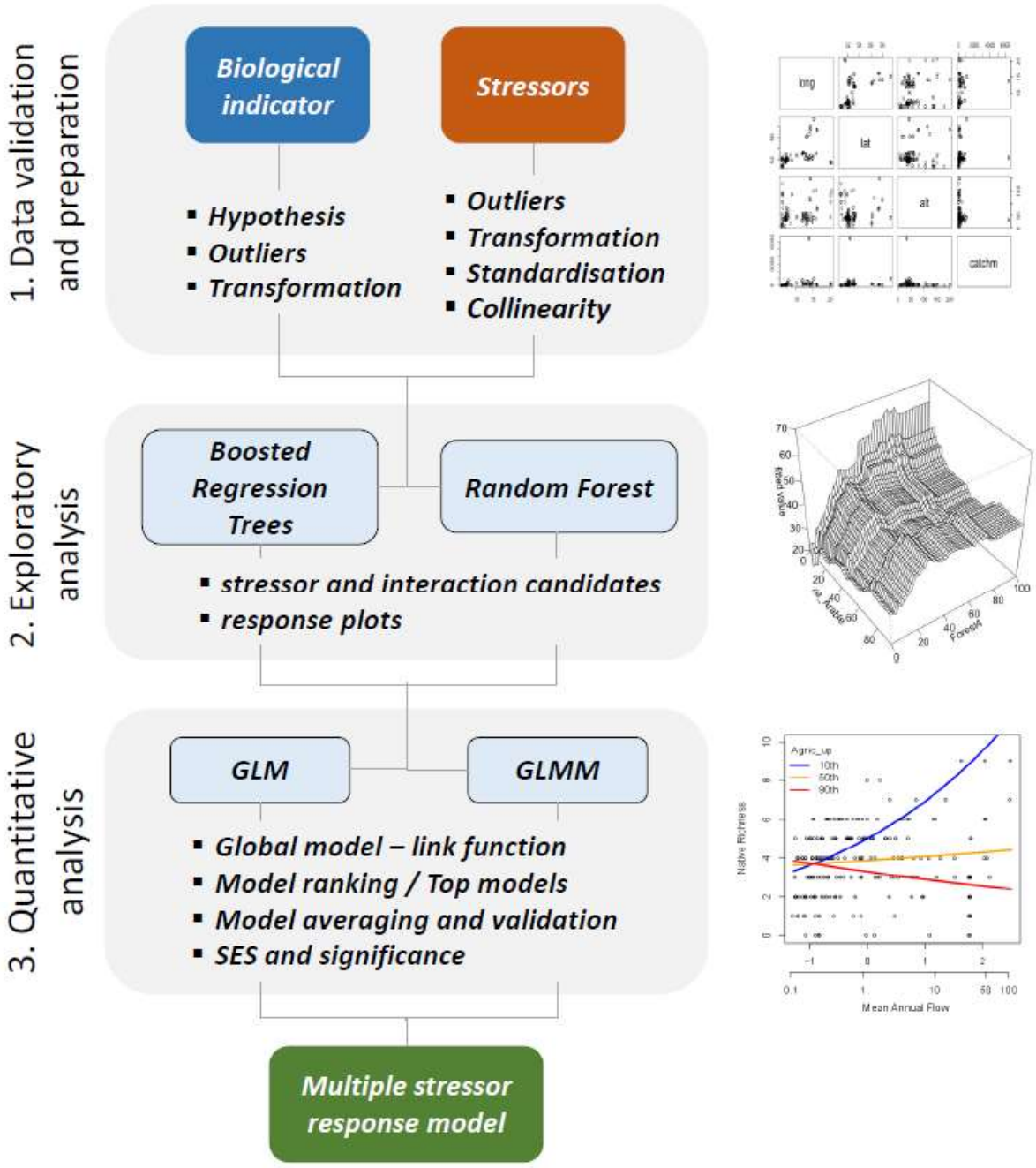



Figure 1: Procedimiento para hacer un análisis de estresores múltiples (extraída de Feld et al., 2016).

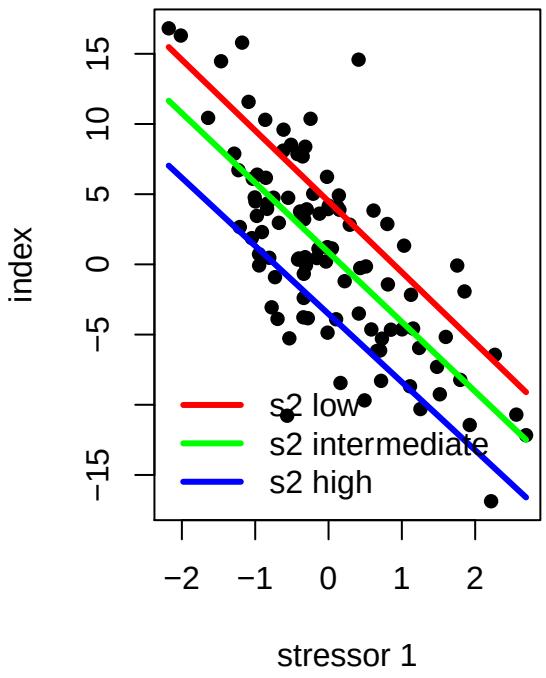
```
 BiocManager::install("variancePartition", version = "3.8")
install.packages("usdm")
install.packages("randomForestSRC")
install.packages("ggRandomForests")
install.packages("gbm")
install.packages("dismo")
install.packages("MuMIn")
install.packages("lattice")
```

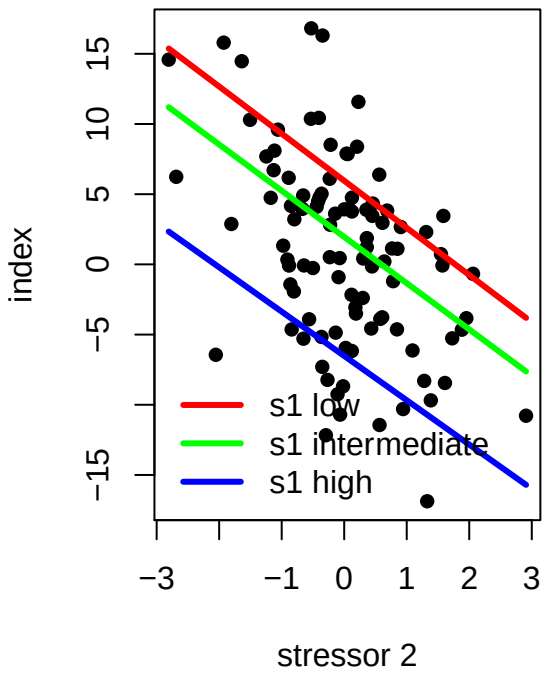
Cargamos las librerías.

```
# Loading required libraries
# The required packages should be installed

library(usdm) # Collinearity
library(randomForestSRC) # RF
library(ggRandomForests) # RF
library(gbm) # BRT
library(dismo) # BRT
library(MuMIn) # Multi-model inference
library(variancePartition)
library(lattice)
```

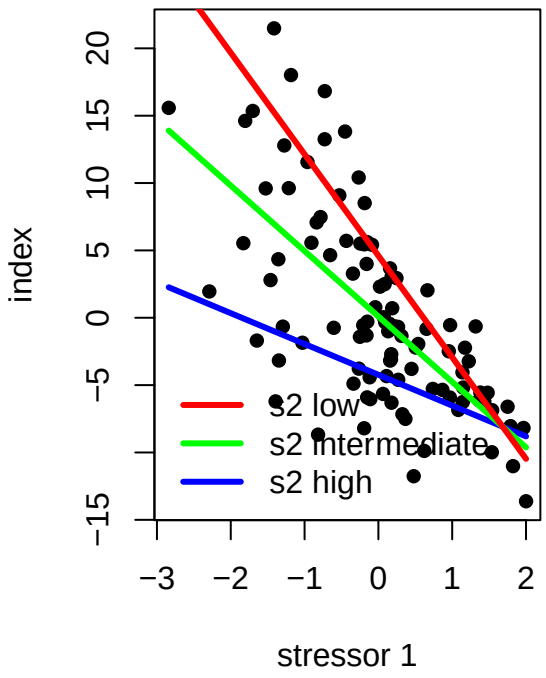
Ahora cargamos las funciones para simular los datos. Fijaros que en este guion utilizamos datos simulados para simplificar este workshop. Sin embargo, puede ser que cuando os enfrenteis a vuestros propios datos, no sea tan fácil. Si es muy complicado, no dudeis en contactar con los instructores para que os den una mano en vuestro trabajo.

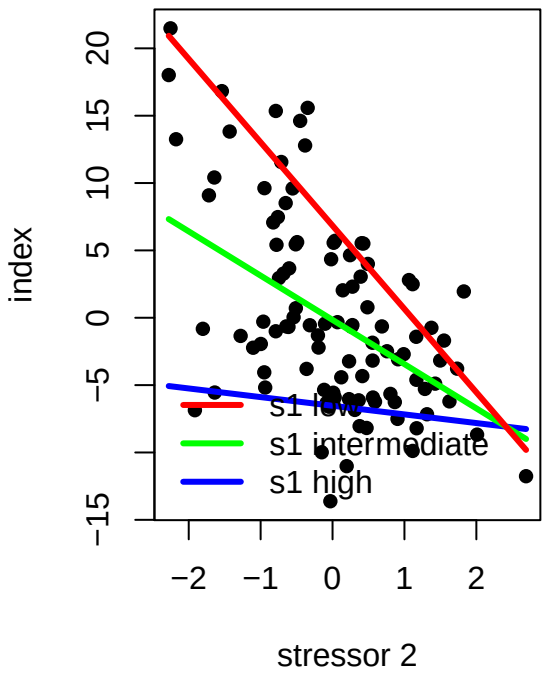
```
# Loading required functions
source("simul_functions.R")

# Simulated dataset

set.seed (1234) # sets a numerical starting point
n <- 100 # number of sites
ac <- 3 # accuracy SD units of error

# Single stressor hierarchy s1 > s2 > s3 = s4
# Interaction hierarchy 1:3 > 1:2 > 2:4
# s1, s2, s3, s4, s1:s2, s1:s3, s2:s4
ses<-c(5, 3, 2, 2, 2, 3, 1)

# Simulating data
sim.multi.str(n, ses, ac, mod.type="additive",plot.int=T)->sim.set.add
```



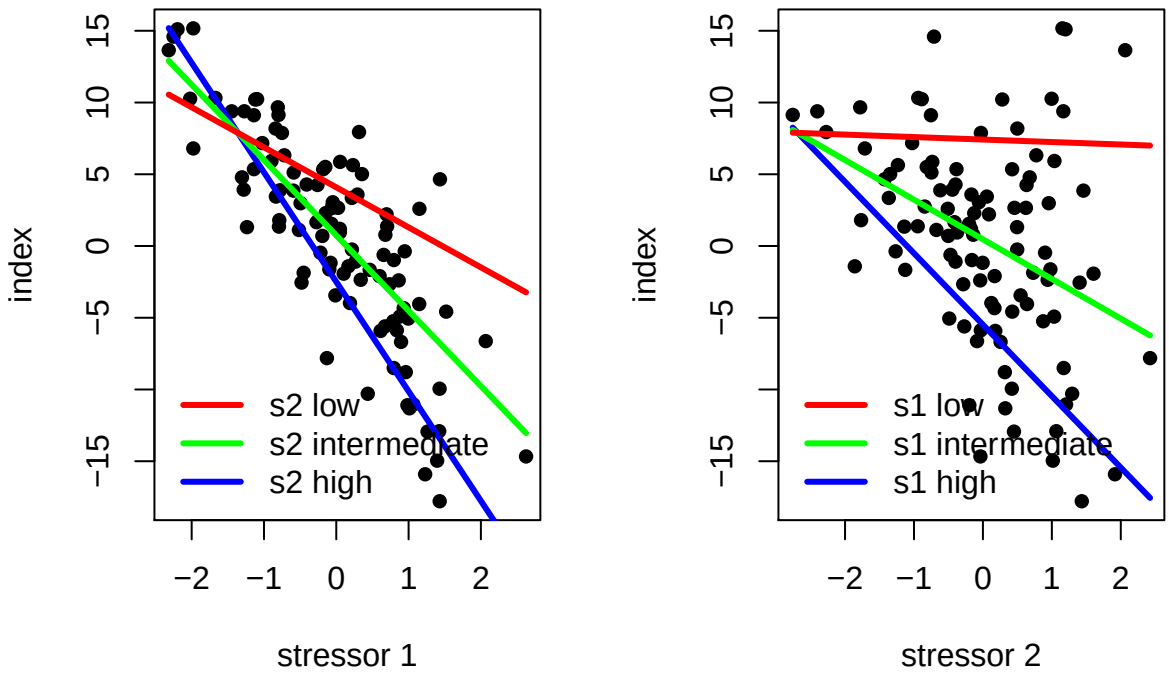




```
sim.multi.str(n, ses, ac, mod.type="antagonistic",plot.int=T)->sim.set.ant
```

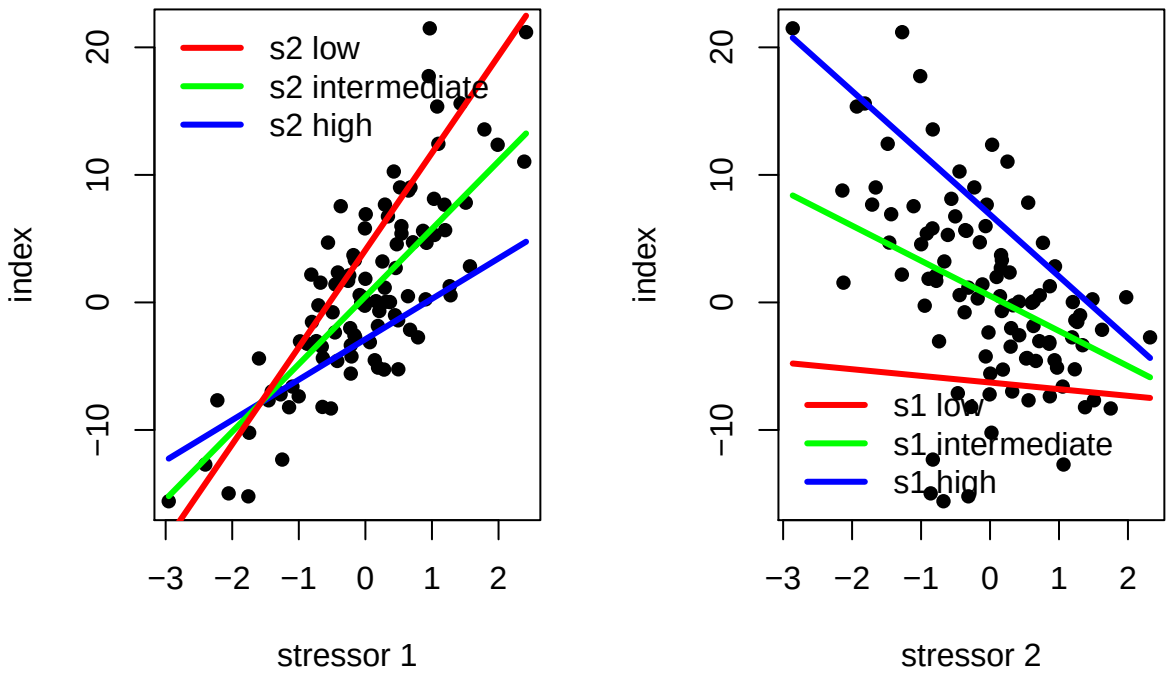



```
sim.multi.str(n, ses, ac, mod.type="synergistic",plot.int=T)->sim.set.syn
```




```
sim.multi.str(n, ses, ac, mod.type="opposing",plot.int=T)->sim.set.opo
```




```
sim.multi.str(n, ses, ac, mod.type="mixed",plot.int=F)->sim.set

sim.set$sim.dat->dat
```

## Preparando los datos

Una vez que tenemos nuestra base de datos a punto y con todas las variables que deseamos investigar, podemos empezar a explorar los datos y prepararlos para el análisis.

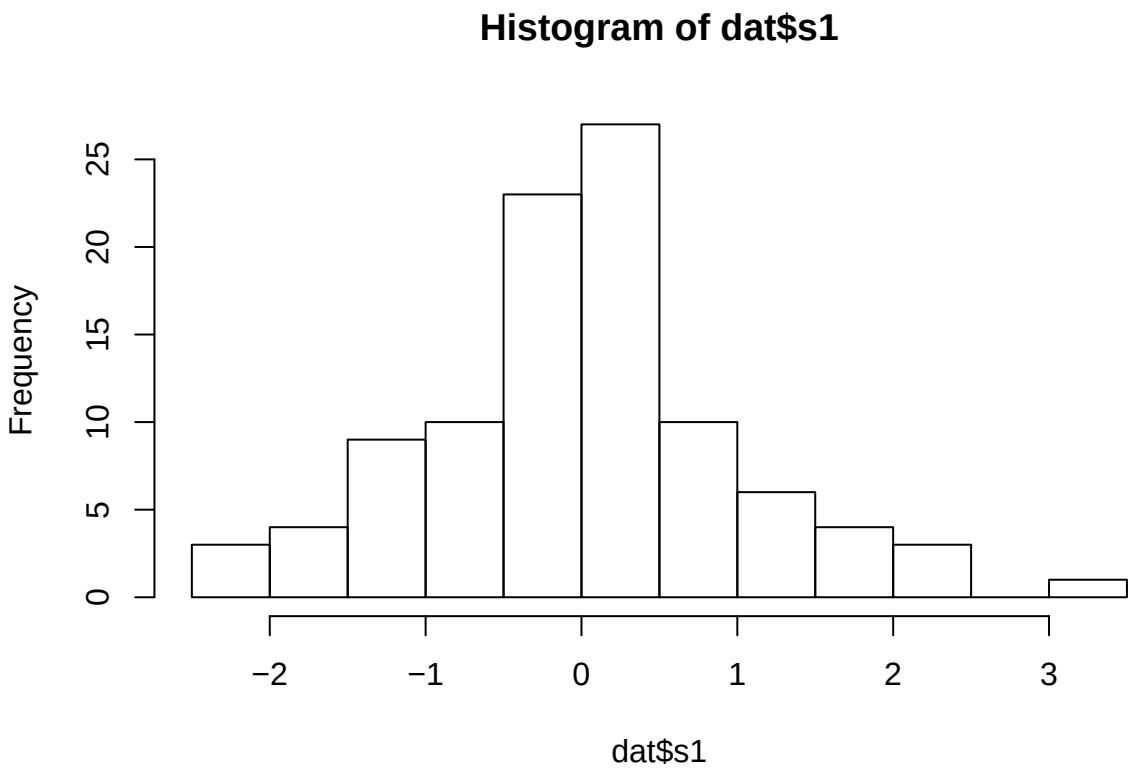
En esta fase, las operaciones fundamentales que vamos a realizar son: 1. Visualizar la distribución de nuestras variables a través de histogramas 2. Transformar aquellas variables que presenten distribuciones sesgadas ("no-acampanadas").

Esto favorecerá la linealidad de las relaciones entre variables y que podamos cumplir las asunciones de normalidad y homocedasticidad de los residuos de nuestros modelos.

Las transformaciones más típicas son las logarítmicas, logit y raíz cuadrada (aunque hay muchas más) y dependerán de la forma de la distribución de los datos originales (sin transformar). Podemos probar varias transformaciones hasta que alcancemos una distribución más acampanada. En muchas ocasiones no se puede obtener una forma apropiada debido a que el número de observaciones es bajo, porque se trata de una distribución bimodal o por otras razones.

Podemos visualizar la distribución de las variables de nuestro set de datos `dat`, usando la función `hist()`.

```
hist(dat$s1) # muestra el histograma de la variable s1
```

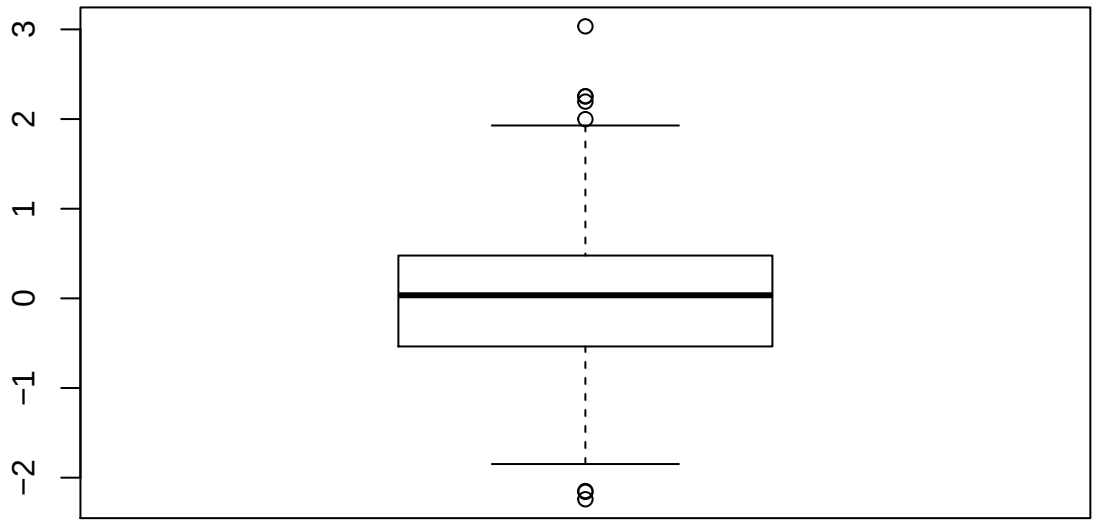



### Outliers

Podemos definir "*outliers*" aquellos valores extremadamente elevados o bajos con respecto a nuestro set de datos.

Una buena forma para de detectar outliers visualmente es usando la función `boxplot()`

```
# outliers
boxplot(dat$s1)
```




## Colinealidad

Los modelos requieren que las variables sean independientes. No afecta a la significación del modelo pero si a sus predicciones. Para ello podemos utilizar las funciones `cor()` (correlación) y/o `vif()` (*variance inflation*).

```
# Collinearity

# calculates pairwise Pearson correlation coefficients for all variables
# of the object dat; note that the function     is only applicable to numerical variables

round(as.dist(cor(dat[,-1])),2)
```

```
##        s1    s2    s3    s4    s5    s6    s7    s8    s9   s10   s11
## s2  -0.08
## s3   0.08  0.13
## s4  -0.01  0.05  0.08
## s5   0.41 -0.14 -0.07 -0.20
## s6   0.63  0.03  0.08  0.00  0.17
## s7  -0.12  0.35 -0.01 -0.03 -0.06 -0.10
## s8   0.02  0.44  0.05  0.05 -0.01  0.07  0.13
## s9   0.09  0.00  0.42 -0.05  0.01  0.01  0.03  0.00
## s10  0.04 -0.03  0.55  0.16 -0.03  0.01 -0.07 -0.13  0.28
## s11  0.03  0.09  0.08  0.40  0.08 -0.01 -0.01  0.11  0.05  0.09
## s12 -0.05  0.17 -0.09  0.54 -0.20  0.14  0.12  0.16 -0.04  0.06  0.32
## s13 -0.10  0.01 -0.11 -0.08 -0.11  0.00  0.07  0.06  0.01 -0.17 -0.08
## s14 -0.03  0.08  0.00  0.03 -0.18  0.08  0.14  0.04  0.30  0.15 -0.03
## s15 -0.12 -0.16 -0.11  0.09 -0.05 -0.02  0.00 -0.14  0.02 -0.20 -0.23
```

```
## s16  0.06 -0.04  0.17  0.02 -0.15 -0.02  0.10  0.11  0.06 -0.01 -0.09
## s17 -0.12 -0.09 -0.10  0.16 -0.06 -0.11  0.03  0.04 -0.09 -0.05  0.23
## s18  0.08  0.09  0.05  0.11 -0.12  0.15 -0.09  0.04  0.06  0.06 -0.01
## s19  0.02 -0.14  0.01 -0.13 -0.03  0.10 -0.04  0.01 -0.09 -0.02  0.07
## s20 -0.03 -0.04 -0.08  0.01 -0.04 -0.01 -0.06  0.00 -0.08  0.14 -0.03
##       s12   s13   s14   s15   s16   s17   s18   s19
## s2
## s3
## s4
## s5
## s6
## s7
## s8
## s9
## s10
## s11
## s12
## s13  0.13
## s14  0.09 -0.05
## s15  0.06  0.12  0.06
## s16  0.03  0.15 -0.04  0.14
## s17  0.05 -0.03 -0.14 -0.08 -0.06
## s18  0.04  0.05  0.01 -0.02 -0.07  0.03
## s19  0.07  0.21  0.04  0.02  0.04  0.06 -0.05
## s20 -0.10  0.03  0.11 -0.06 -0.08 -0.19  0.00  0.10
```

```
# Exploring collinearity using Variance Inflation Factor
vif(dat[,-1])
```

```
##    Variables      VIF
## 1         s1 2.202518
## 2         s2 1.640392
## 3         s3 2.173731
## 4         s4 2.008397
## 5         s5 1.522914
## 6         s6 1.945280
## 7         s7 1.258361
## 8         s8 1.382848
## 9         s9 1.535945
## 10       s10 1.883552
## 11       s11 1.520976
## 12       s12 1.939066
## 13       s13 1.208729
## 14       s14 1.349511
## 15       s15 1.313641
## 16       s16 1.220853
## 17       s17 1.255370
## 18       s18 1.108376
## 19       s19 1.249364
## 20       s20 1.211903
```

En esta fase no hace falta que tomemos decisiones sobre si debemos eliminar variables altamente correlacionadas o valores extremos, pero esta información nos vendrá bien para refinar los modelos que realicemos en las siguientes fases. En algunos casos, los valores extremos podrían alterar los modelos tipo CART o por supuesto a los modelos de tipo GLM/GLMM. Más adelante, cuando seleccionemos las variables más importantes, sí

que estudiemos qué variables o casos tendremos que eliminar.

# Explorando los datos

Una vez que los datos están preparados, pasaremos a explorar qué variables predictoras son más importantes. Para esta tarea, podemos usar diversas técnicas, pero en este caso nos centraremos en tres:

- Correlaciones
- Random Forest
- Boosted Regression Trees.

Estas herramientas exploratorias son potentes pero debemos usarlas de manera racional y siguiendo buenos criterios de modelado (ver sección Lecturas Recomendadas para ver limitaciones y críticas a los CART).

Si tenemos datos con estructuras de dependencia espacial (estructuras anidadas, medidas repetidas) y/o temporal (series temporales), tendremos que ser extremadamente cuidadosos a la hora de ejecutar los análisis exploratorios. En algunos casos, podremos solucionar este problema seleccionando subconjuntos de datos para los cuales no tengamos medidas repetidas o dependientes. También podemos incluir la variable que da cuenta de la estructura anidada o de autocorrelación temporal de nuestros datos, aunque esto puede entrañar cierto riesgo. Ver Appendix 3 en Feld et al. (2016) para más detalles y posibles soluciones.

## Correlaciones

Las correlaciones nos muestran, a través de un coeficiente, la relación que tienen dos variables. Los coeficientes pueden tomar valores de -1 hasta +1. Los coeficientes negativos implican que el incremento de una variable está relacionado con el descenso de otra. Los coeficientes positivos indican que ambas variables tienen tendencias muy similares. Un coeficiente próximo a 0 indica que ambas variables tienen poca relación.

Podemos usar dos tipos de correlación: a) el coeficiente de correlación de Pearson (r) mide el grado de relación lineal entre dos variables; b) el coeficiente de Spearman ($\rho$), que mide la asociación entre dos variables, pero es menos sensible a valores extremos y falta de linealidad.

```
cor(dat$y, dat$s1, method="pearson") # correlation between y, s1
```

```
## [1] -0.4976722
```

```
cor(dat$y,dat$s1, method="spearman") # correlation between y, s1
```

```
## [1] -0.5711371
```

## Random Forest

Los Random Forest (RF) son técnicas de modelado no-paramétricas con una alta flexibilidad para manejar variables de naturaleza diversa. Así, las variables (respuesta y predictores) pueden ser de tipo continuo, discreto, categórico y binario. Este tipo de modelos también incluye la posibilidad de usar variables predictoras con valores ausentes, conocidos como "*missing values*" o NAs. Además, los RF pueden modelar datasets con pocas observaciones y muchos predictores. También tienen capacidad para modelar respuestas no lineales e interacciones (Breiman, 2001; Ishwaran et al., 2014).

Simplificando, los RF ejecutan una serie de modelos (árboles de decisión, *regression trees*) basados en subconjuntos de nuestro *dataset* ("in the bag"=learning data), que normalmente suponen dos tercios de las observaciones de las que disponemos. Los modelos dividen nuestros datos según criterios binarios tales como pH<7, altura > 200 m o color == "amarillo". Cada uno de estos modelos se aplica al tercio restante de observaciones ("out-of-bag"=test data) para identificar las variables predictoras más importantes y la capacidad predictiva del modelo. Finalmente, todos estos modelos se combinan para producir un modelo final más potente. Para más información sobre los Random Forest y es recomendable revisar literatura especializada (Breiman, 2001; Cutler et al., 2007; Ishwaran et al., 2014; Strobl et al., 2009, 2008).

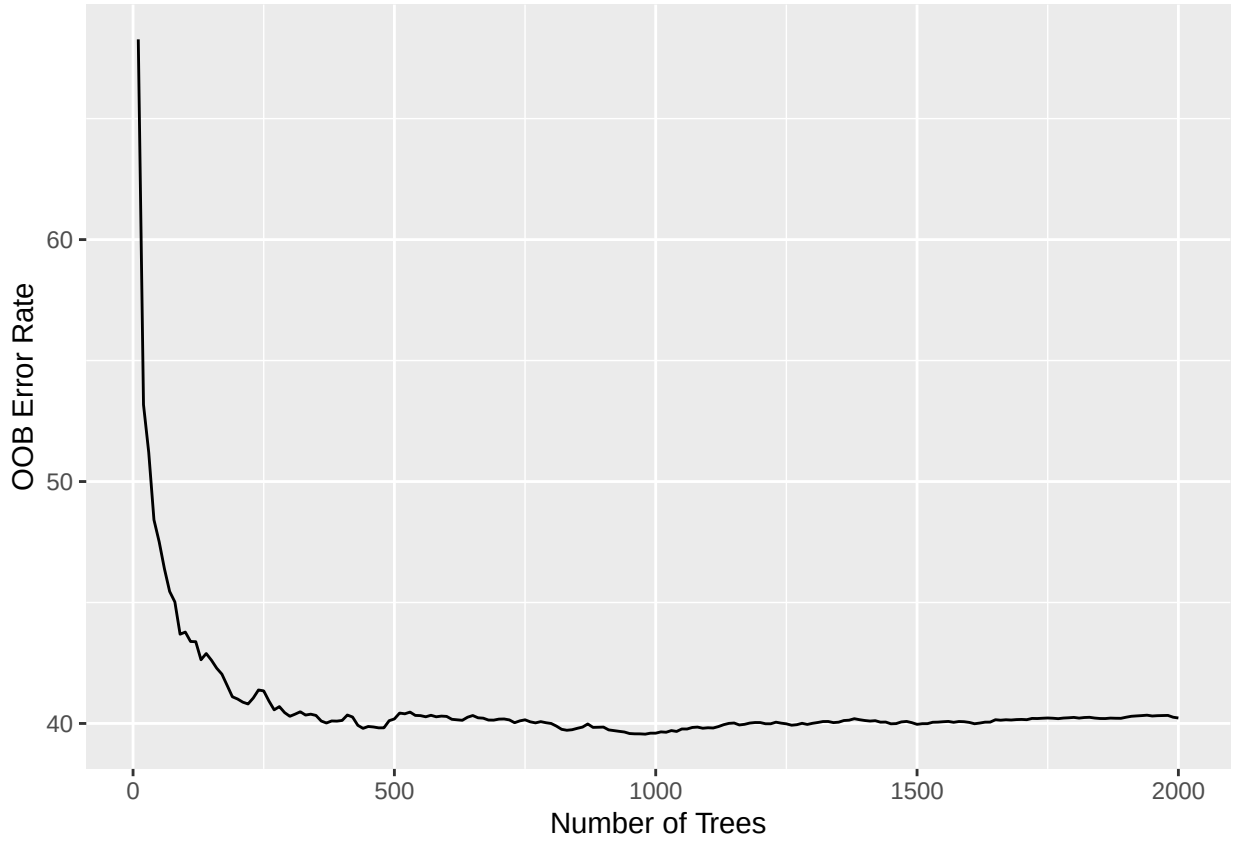
En este taller vamos a usar la función `rfsrc()` de la libería `randomForestSRC` (Ishwaran et al., 2014). En esta función tenemos que introducir:

- `model formula`: las variables del modelo
- `mtry`: número de predictores usados para cada división binaria en los árboles de decisión. Normalmente suele ser un tercio del total de predictores si estamos haciendo una regresión.
- `ntree`: número de árboles (número de modelos a ejecutar)
- `importance`: método de cálculo de la importancia de los predictores

```
# Función para ejecutar el RF
# my.rf nos da los detalles del modelo , e.g. bondad de ajuste, out-of-bag (OBB) error
my.rf <- rfsrc (y ~ ., mtry = 6, ntree = 2000, importance = "permute",  data = dat)
my.rf
```

```
##                          Sample size: 100
##                       Number of trees: 2000
##             Forest terminal node size: 5
##         Average no. of terminal nodes: 20.441
## No. of variables tried at each split: 6
##                 Total no. of variables: 20
##         Resampling used to grow trees: swr
##      Resample size used to grow trees: 100
##                              Analysis: RF-R
##                                Family: regr
##                        Splitting rule: mse *random*
##         Number of random split points: 10
##                   % variance explained: 38.28
##                             Error rate: 40.23
```

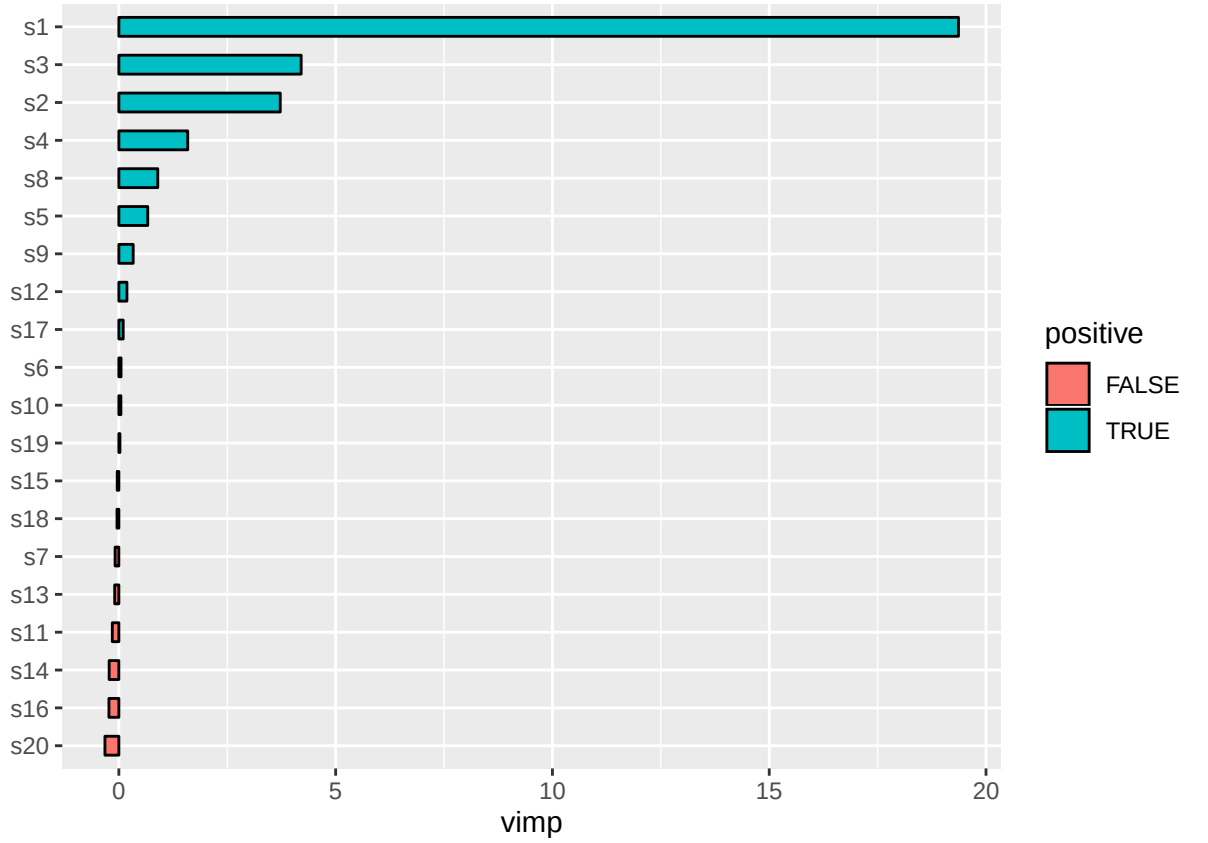
```
# plot para determinar el número óptimo de árboles
plot (na.omit(gg_error (my.rf)))
```




```
# gg_vimp() importancia de los predictores
my.rf.vimp <- gg_vimp (my.rf)
my.rf.vimp
```

```
##    vars  set        vimp positive
## 1    s1 VIMP 19.36432065     TRUE
## 2    s3 VIMP  4.20455076     TRUE
## 3    s2 VIMP  3.72227362     TRUE
## 4    s4 VIMP  1.58829634     TRUE
## 5    s8 VIMP  0.89727414     TRUE
## 6    s5 VIMP  0.66627051     TRUE
## 7    s9 VIMP  0.33036231     TRUE
## 8   s12 VIMP  0.18598303     TRUE
## 9   s17 VIMP  0.09983943     TRUE
## 10   s6 VIMP  0.05044155     TRUE
## 11  s10 VIMP  0.04688725     TRUE
## 12  s19 VIMP  0.02433822     TRUE
## 13  s15 VIMP -0.03735242    FALSE
## 14  s18 VIMP -0.04292053    FALSE
## 15   s7 VIMP -0.08632849    FALSE
## 16  s13 VIMP -0.09580002    FALSE
## 17  s11 VIMP -0.15137070    FALSE
## 18  s14 VIMP -0.22352596    FALSE
## 19  s16 VIMP -0.22880235    FALSE
## 20  s20 VIMP -0.32090299    FALSE
```

```r
plot (my.rf.vimp) # plot mostrando la importancia de los predictores
```




```r
# Variables más importantes
md.obj <- max.subtree (my.rf)
md.obj$topvars
```

```
## [1] "s1" "s2" "s3" "s4"
```

```r
# Explorando interacciones
my.rf.interact<-find.interaction(my.rf, xvar.names = md.obj$topvars,
                                 importance= "permute", method = "vimp",
                                 nrep = 3)
```

```
## Pairing s1 with s3
## Pairing s1 with s2
## Pairing s1 with s4
## Pairing s3 with s2
## Pairing s3 with s4
## Pairing s2 with s4
##
##                                Method: vimp
##                     No. of variables: 4
##            Variables sorted by VIMP?: TRUE
##    No. of variables used for pairing: 4
##     Total no. of paired interactions: 6
##              Monte Carlo replications: 3
##     Type of noising up used for VIMP: permute
```

```
##
##         Var 1  Var 2  Paired Additive Difference
## s1:s3 24.6830 3.9641 25.9621  28.6471    -2.6849
## s1:s2 24.6830 4.1408 28.4797  28.8238    -0.3441
## s1:s4 24.6830 1.9757 26.9223  26.6587     0.2636
## s3:s2  3.9921 4.1408  8.2485   8.1329     0.1156
## s3:s4  3.9921 1.9757  5.8704   5.9678    -0.0974
## s2:s4  4.1801 1.9757  6.3342   6.1558     0.1784
```

## Boosted Regression Trees

Los Boosted Regression Tree analysis (BRT) tienen unas características muy similares a los RF, aunque usan algoritmos distintos para producir los modelos finales. Recomendamos la lectura de Elith et al. (2008) para más detalles.

En este taller vamos a usar la función `gbm.step` y las liberías `gbm` (Ridgeway, 2015) y `dismo` (Hijmans R.J. et al., 2016). En esta función tenemos que introducir:

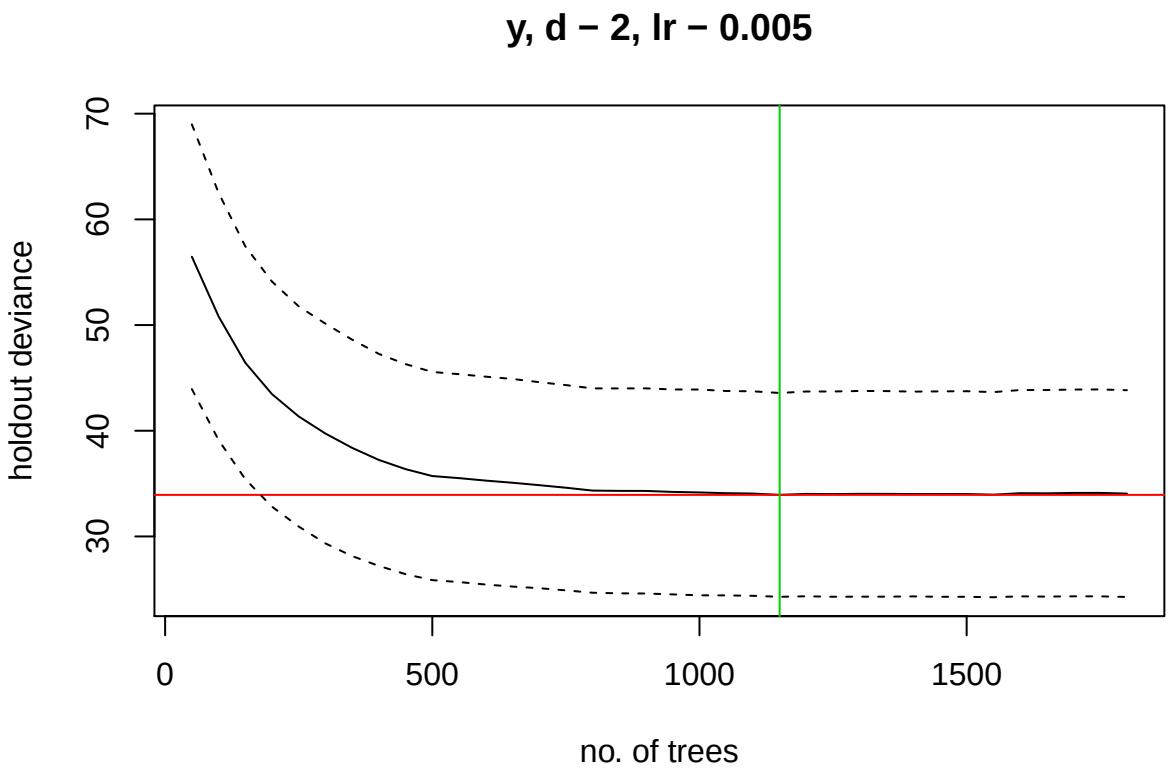
- `gbm.y`: variable respuesta (número de columna)
- `gbm.x`: predictores (número de columna)
- `family`: define el tipo de función y el error de la distribución para la variable respuesta. Usaremos "gaussian" para variables contínuas, "poisson" para variables discretas (e.g. number of species) y "bernoulli" para variables binomiales (0, 1).
- `learning.rate` determina el número total de modelos ejecutados (trees). Valores pequenos tienden a producir muchos modelos, mientras que valores altos producirán menos modelos. Podemos empezar a probar con learning.rate=0.005 e ir aumentando dicho valor.
- `bag.fraction`: porcentaje de observaciones que observaciones que usaremos para hacer cada modelo (árbol)
- `tree.complexity`: define el tipo de interacciones que se van a testar. Valores de 1 indican un modelo puramente aditivo. Para testar interacciones entre pares de variables tendremos que establecer un valor de 2.

Para determinar los valores más apropiados de `learning.rate` debemos ejecutar el modelo varias veces con valores distintos de estos dos parámetros hasta que encontremos valores estables. Se recomienda que el modelo BRT que construyamos tenga al menos 1000 árboles (*regression trees*) (Elith et al. 2008).

```
# BRT con interacciones entre parejas de predictores y un 67% de los datos
#usados para construir el modelo
my.brt <- gbm.step (data = dat, gbm.x = c(2:ncol(dat)), gbm.y = 1,
                    family = "gaussian", tree.complexity = 2,
                    learning.rate = 0.005,  bag.fraction = 0.67)
```

```
##
##
##  GBM STEP - version 2.9
##
## Performing cross-validation optimisation of a boosted regression tree model
## for y and using a family of gaussian
## Using 100 observations and 20 predictors
## creating 10 initial models of 50 trees
##
##  folds are unstratified
## total mean deviance =  64.5225
## tolerance is fixed at  0.0645
## ntrees resid. dev.
## 50    56.4665
```

```
## now adding trees...
## 100   50.8201
## 150   46.4388
## 200   43.4655
## 250   41.3528
## 300   39.7358
## 350   38.3813
## 400   37.2408
## 450   36.3683
## 500   35.7141
## 550   35.5179
## 600   35.2832
## 650   35.0801
## 700   34.8504
## 750   34.6089
## 800   34.3398
## 850   34.3106
## 900   34.3049
## 950   34.2122
## 1000   34.1629
## 1050   34.0913
## 1100   34.057
## 1150   33.9327
## 1200   34.0253
## 1250   34.0046
## 1300   34.0317
## 1350   34.0267
## 1400   34.0143
## 1450   34.0075
## 1500   34.0143
## 1550   33.9506
## 1600   34.0923
## 1650   34.0797
## 1700   34.1168
## 1750   34.1198
## 1800   34.0533

## fitting final gbm model with a fixed number of 1150 trees for y
```

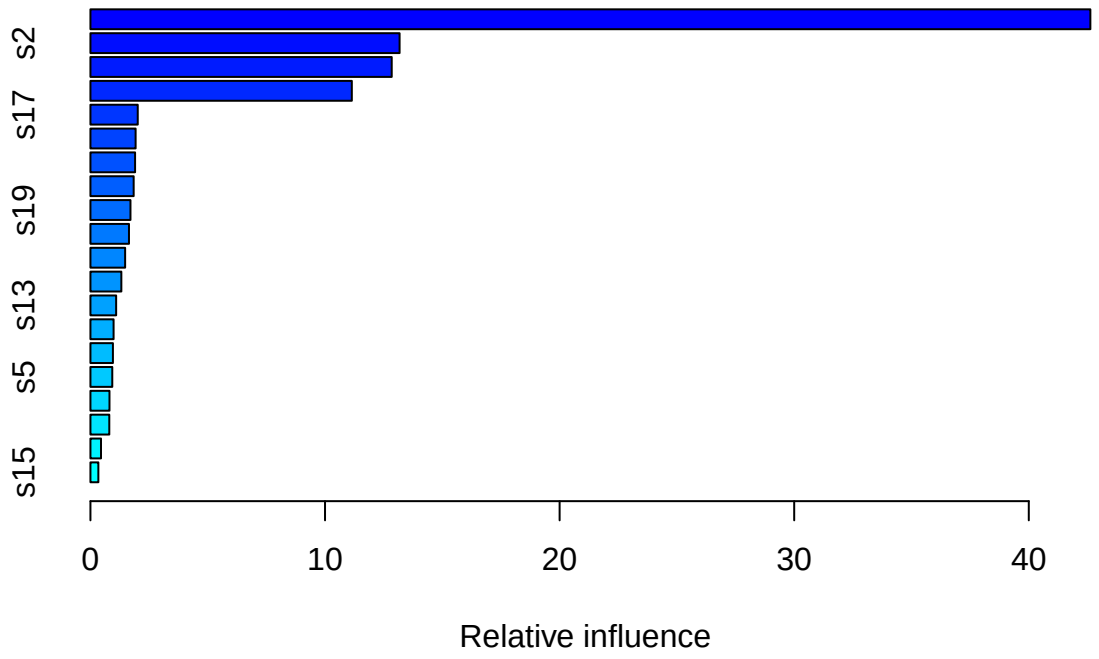



```
##
## mean total deviance = 64.522
## mean residual deviance = 9.578
##
## estimated cv deviance = 33.933 ; se = 9.643
##
## training data correlation = 0.938
## cv correlation =  0.689 ; se = 0.111
##
## elapsed time -  0.04 minutes
```

```
# Bondad de ajuste del modelo basada en el cross validation
my.brt$self.statistics$mean.null -> null.dev
my.brt$cv.statistics$deviance.mean -> resid.dev
1-resid.dev/null.dev
```

```
## [1] 0.4740956
```

```
# Importancia de los predictores
brt.imp <- summary (my.brt)
```

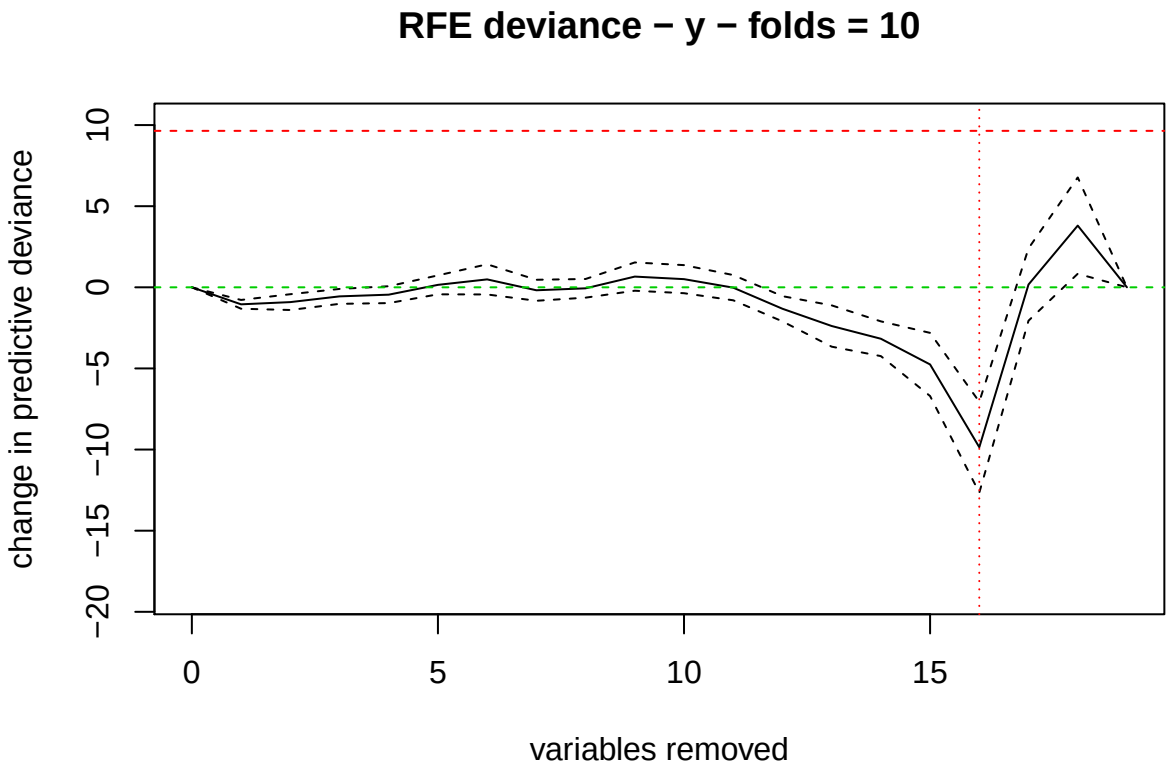



```
brt.imp
```

```
##     var    rel.inf
## s1   s1 42.6248802
## s2   s2 13.1788702
## s3   s3 12.8430002
## s4   s4 11.1426622
## s17 s17  2.0168880
## s14 s14  1.9259617
## s8   s8  1.9029990
## s9   s9  1.8431599
## s19 s19  1.7064131
## s18 s18  1.6426019
## s12 s12  1.4790670
## s6   s6  1.3175876
## s13 s13  1.0957245
## s7   s7  0.9872608
## s20 s20  0.9584817
## s5   s5  0.9308262
## s10 s10  0.8119698
## s11 s11  0.8041551
## s16 s16  0.4499519
## s15 s15  0.3375389
```

```
# Variables más importantes
my.brt.simp <- gbm.simplify (my.brt)
```

```
## gbm.simplify - version 2.9
## simplifying gbm.step model for y with 20 predictors and 100 observations
## original deviance = 33.9327(9.6434)

## variable removal will proceed until average change exceeds the original se

## creating initial models...

## dropping predictor: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
## processing final dropping of variables with full data
## 1-s15
## 2-s16
## 3-s11
## 4-s20
## 5-s5
## 6-s10
## 7-s7
## 8-s6
## 9-s13
## 10-s9
## 11-s18
## 12-s19
## 13-s17
## 14-s8
## 15-s12
## 16-s14
## 17-s3
## 18-s4
```



```
# Explorando interacciones
int.my.brt <- gbm.interactions (my.brt) # objeto con los resultados
```

```
## gbm.interactions - version 2.9
## Cross tabulating interactions for gbm model with 20 predictors
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

```
int.my.brt$interactions # interacciones
```

```
##     s1    s2      s3     s4   s5   s6   s7    s8    s9  s10  s11  s12  s13
## s1   0 54.91 1274.47 100.78 0.00 1.99 0.04 13.69 21.33 1.09 0.03 0.01 0.01
## s2   0  0.00    0.05  14.52 2.39 0.01 0.01  0.00  0.42 0.23 0.59 1.31 2.42
## s3   0  0.00    0.00   0.90 0.01 3.21 0.03  0.01  0.02 0.93 0.00 0.03 0.00
## s4   0  0.00    0.00   0.00 0.02 0.15 0.12  0.71  0.22 0.00 0.00 1.18 3.82
## s5   0  0.00    0.00   0.00 0.00 0.00 0.33  0.00  0.06 0.02 0.01 0.00 0.00
## s6   0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.01 0.12 0.06 0.02 0.01
## s7   0  0.00    0.00   0.00 0.00 0.00 0.00  0.02  0.02 0.01 0.08 0.01 0.00
## s8   0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.03 0.00 0.00 0.00 0.23
## s9   0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.01 0.00 0.00 0.01
## s10  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.01 0.00
## s11  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.11 0.00
## s12  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.20
## s13  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s14  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s15  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s16  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s17  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s18  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s19  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
## s20  0  0.00    0.00   0.00 0.00 0.00 0.00  0.00  0.00 0.00 0.00 0.00 0.00
##      s14  s15  s16  s17  s18  s19  s20
## s1  0.22 0.03 0.02 0.18 0.74 0.10 0.66
## s2  1.06 0.01 0.00 9.50 0.21 4.63 0.09
## s3  0.10 0.01 0.01 0.01 1.04 0.25 0.02
## s4  0.97 0.00 0.07 0.48 0.67 3.12 0.13
## s5  0.00 0.02 0.00 0.00 0.00 0.00 0.00
## s6  0.06 0.08 0.00 0.27 0.01 0.12 0.00
## s7  0.00 0.00 0.13 0.17 0.15 0.08 0.01
## s8  0.00 0.00 0.08 0.26 0.01 0.13 0.07
## s9  0.19 0.01 0.00 0.00 0.02 0.26 0.00
## s10 0.18 0.01 0.00 0.00 0.00 0.01 0.00
## s11 0.11 0.02 0.00 0.00 0.00 0.01 0.12
## s12 0.00 0.01 0.03 0.09 0.02 0.04 0.00
## s13 0.55 0.00 0.00 0.00 0.00 0.02 0.03
## s14 0.00 0.18 0.24 0.03 0.54 0.03 0.01
## s15 0.00 0.00 0.00 0.00 0.01 0.00 0.00
## s16 0.00 0.00 0.00 0.01 0.01 0.26 0.01
## s17 0.00 0.00 0.00 0.00 0.21 0.06 0.02
## s18 0.00 0.00 0.00 0.00 0.00 0.00 0.02
## s19 0.00 0.00 0.00 0.00 0.00 0.00 0.01
## s20 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
int.my.brt$rank.list # lista con el ranking de las interacciones
```

```
##    var1.index var1.names var2.index var2.names int.size
```

```
##    1          3         s3          1         s1  1274.47
##    2          4         s4          1         s1   100.78
##    3          2         s2          1         s1    54.91
##    4          9         s9          1         s1    21.33
##    5          4         s4          2         s2    14.52
##    6          8         s8          1         s1    13.69
##    7         17        s17          2         s2     9.50
##    8         19        s19          2         s2     4.63
##    9         13        s13          4         s4     3.82
##   10          6         s6          3         s3     3.21
##   11         19        s19          4         s4     3.12
##   12         13        s13          2         s2     2.42
##   13          5         s5          2         s2     2.39
##   14          6         s6          1         s1     1.99
##   15         12        s12          2         s2     1.31
##   16         12        s12          4         s4     1.18
##   17         10        s10          1         s1     1.09
##   18         14        s14          2         s2     1.06
##   19         18        s18          3         s3     1.04
##   20         14        s14          4         s4     0.97
```

# Modelos finales

## Selección de predictores

Una vez hemos ejecutado los modelos RF y/o BRT nos toca elegir qué predictores vamos a usar. Para ello nos basaremos en su importancia relativa y también en información teórica o empírica previa, como hemos comentado al principio (Crawley, 2014; Grueber et al., 2011). Existen algunas funciones de R que hacen una selección automática de los predictores más importantes para RF y BRT. Sin embargo, esta no es la opción más recomendada, ya que esta selección tiene que ser supervisada para que nos aseguremos de que no se escapa ninguna variable clave y de que estas variables tienen un sentido biológico/ecológico. Es probable que RF y BRT nos den resultados ligeramente distintos, sobre todo si el número de observaciones no es grande (e.g. $>$ 100 observaciones). En estos casos, observaremos qué variables son importantes en ambos modelos y cuáles difieren. Para aquellas variables que muestren grandes diferencias entre RF y BRT, tendremos que tomar una decisión basada en nuestra información previa y su sentido ecológico.

La cantidad de predictores que seleccionemos está condicionada por el número de observaciones que tengamos. De manera general, podemos seguir la regla de *one in ten*, que implica un término predictor por cada 10 observaciones. Así, si modelamos un dataset con 72 observaciones, podremos incluir hasta siete términos en el modelo. En algunos casos estas reglas pueden cambiar, pero es importante ser conservador para asegurar unos resultados robustos.

A la hora de elegir los predictores, debemos evitar incluir aquellos que tengan una alta colinealidad. Podemos usar como criterio aquellas parejas de predictores con una correlación de Pearson r $\geq$ 0.70, de las que eliminaremos el predictor que tenga una menor importancia predictiva en los RF o BRT y/o aquel que tenga menos sentido biológico/ecológico. Finalmente, es conveniente que estandaricemos los predictores usando la función `scale()`, que los convierte en variables con `media=0` y `SD=1`. Esto nos permitirá una correcta estimación y comparación del tamaño del efecto de los predictores (coeficientes de regresión), ya que estarán en las mismas unidades. Si tenemos variables cualitativas tendremos que aplicar otro tipo de estandarizaciones (Grueber et al., 2011; ver también sum-to-zero contrasts).

## Selección de modelos: métodos tradicionales vs. multi-model inference

De manera tradicional, en ecología, se han usado procedimientos *step-wise* para averiguar qué predictores son más importantes a la hora de predecir nuestra variable de interés (Grueber et al., 2011; Johnson and Omland,

2004). Este procedimiento de selección de modelos consiste en ir añadiendo (*forward selection*) o quitando (*backward selection*) variables predictoras en función de su capacidad para explicar nuestros datos y de la complejidad del modelo. Existen varias medidas para medir la bondad de ajuste y la complejidad del modelo. Entre los más utilizados se encuentran el Akaike Information Criterion (AIC) y su versión para muestras pequeñas (observaciones/parámetros estimados $< 40$; AICc). En este taller usaremos AIC por cuestiones de simplicidad.

Sin embargo, esta estrategia conlleva limitaciones importantes como, por ejemplo, una capacidad reducida para testar varias hipótesis al mismo tiempo, el riesgo de excluir variables explicativas importantes que no aparezcan en el modelo elegido o la obtención de tamaños de efecto sesgados para algunas variables explicativas (Johnson and Omland, 2004).

En los últimos años, se han empezado a usar metodologías basadas en la selección e inferencia a partir de varios modelos, conocidas como *model selection* y *multi-model inference* (Burnham and Anderson, 2002; Grueber et al., 2011; Johnson and Omland, 2004), que son capaces de solventar las limitaciones de los métodos *step-wise*. Este enfoque se basa en promediar los coeficientes de los mejores modelos que incluyan distintas combinaciones de los predictores más importantes. Así, en vez de seleccionar un solo modelo que minimice el AIC, vamos a seleccionar un conjunto de modelos que cumplan una serie de criterios (e.g. aquellos que no difieran en más de 2, 4 o 6 unidades de AIC respecto al modelo con el AIC más bajo o aquellos modelos que acumulen un 95%). Con esta metodología, obtendremos un modelo final que recoge información de un subconjunto grande de modelos y permite testar varias hipótesis al mismo tiempo.

Aunque proporcionaremos recomendaciones para su correcta aplicación, esta estrategia tampoco está exenta de críticas ni limitaciones (Cade, 2015; Tyre, 2017; Walker, 2018).

## Multi-model inference

Una vez que hemos elegido todas las variables que van a formar parte del modelo, calcularemos los coeficientes (tamaño del efecto) y el p-valor de los predictores. Para ello, seguiremos tres pasos: 1) Modelo global, 2) Multi-model inference y 3) validación de las asunciones del modelo

1. **Establecer el modelo global y definir la función de enlace**

Para construir el modelo global usaremos un tipo de modelo que sea apropiado en relación a la estructura de nuestros datos. Este modelo incluirá la variable respuesta a modelar y todos los predictores seleccionados con la metodología anteriormente explicada (recordemos la necesidad de incluir solo variables explicativas con un sentido ecológico/biológico claro). Entre las técnicas de modelado disponibles, los GLM ofrecen una gran flexibilidad para modelar variables respuesta de tipo continuo, discreto y binario que no tengan estructuras de dependencia espacial o temporal (e.g. una medida por localidad, y que las localidades sean independientes). Los modelos que tengan estructuras de dependencia espacial y/o temporal se tendrán que modelar usando modelos de tipo GLMM o GAMM, que permiten la inclusión de factores aleatorios ("random incercepts/slopes"). Por simplicidad vamos a ilustrar este ejemplo con un GLM con un error gaussiano. Para más detalles sobre GLM, GAM, GLMM y GAMM recomendamos consultar Zuur et al. (2009)., el GLMM FAQ y el blog From the bottom of the heap.

Una vez hemos decidido qué técnica estadística vamos a usar, crearemos un modelo global con la variable respuesta y todos los predictores de interés respetando la regla de *one in ten*. Entre los predictores, incluiremos términos individuales y las interacciones más relevantes (y términos cuadráticos, si procede). Es muy importante destacar que debemos introducir las interacciones de forma muy cuidadosa y en función de nuestras hipótesis, información previa y también basado en los resultados de los modelos RF y BRT. En R podemos especificar interacciones de dos formas `a:b` o `a*b` (equivalente `a: a+b+a:b`).

```
# Modelo global, función genérica del GLM gaussiano, lm()
mod <- lm (y ~ s1 + s2 + s3 + s6 + s9 + s1:s2 + s1:s3, data=dat)
summary(mod)
```

```
##
## Call:
```

```
## lm(formula = y ~ s1 + s2 + s3 + s6 + s9 + s1:s2 + s1:s3, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9645 -2.8845  0.0743  2.3679 10.9819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7752     0.3968  -1.954  0.05378 .
## s1           -6.0959     0.5466 -11.153  < 2e-16 ***
## s2           -2.7104     0.4049  -6.695 1.67e-09 ***
## s3           -1.3302     0.4432  -3.001  0.00346 **
## s6            0.3021     0.5152   0.586  0.55908
## s9           -0.8771     0.4419  -1.985  0.05011 .
## s1:s2        -1.9080     0.3705  -5.150 1.48e-06 ***
## s1:s3         4.3573     0.4039  10.788  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.943 on 92 degrees of freedom
## Multiple R-squared:  0.7783, Adjusted R-squared:  0.7614
## F-statistic: 46.14 on 7 and 92 DF,  p-value: < 2.2e-16
```

```
r.squaredGLMM(mod)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help
## page.
```

```
##            R2m       R2c
## [1,] 0.7653955 0.7653955
```

2. **Multi-model inference**

En el segundo paso, usaremos la función `dredge()` de la librería `MuMIn` (Barton, 2016) para generar todos los modelos posibles a partir de distintas combinaciones de las variables explicativas contenidas en el modelo global. Recomendamos inspeccionar cuidadosamente los resultados producidos por la función `dredge()`. En esta inspección evaluaremos que los modelos con AIC más bajo tienen sentido ecológico. Además, evaluaremos el rango de variación y la distribución de los tamaños de efecto (coeficientes de regresión) para cada variable explicativa, en particular en aquellos modelos que difieran en 2 unidades con el modelo con el AIC mínimo. Seguidamente, ordenaremos los modelos en función de su AIC u otra medida análoga apropiada (Grueber et al., 2011; Johnson and Omland, 2004), y de su peso relativo de evidencia (Akaike weight). El "weight" nos indica la probabilidad de que un modelo sea el mejor modelo aproximado en relación a nuestros datos, por lo que la suma del weight de todos los modelos siempre sumará 1. En el caso de que un modelo tenga un weight mayor que 0,90 bastaría para seleccionarlo como modelo final y no sería necesario realizar los pasos siguientes (selección de modelos y promedio de sus coeficientes).

```
# ejecuta todos los modelos posibles y clasifica los modelos en función del AIC de cada modelo.
#También se visualiza la bondad de ajuste
options(na.action = "na.fail") # necesario para ejecutar dredge()
mod_d <- dredge (mod, rank = "AIC",
                      extra = c(R2=function(x) r.squaredGLMM(x)))
```

```
## Fixed term is "(Intercept)"
```

```
mod_d # ranking de los modelos producidos
```

```
## Global model call: lm(formula = y ~ s1 + s2 + s3 + s6 + s9 + s1:s2 + s1:s3, data = dat)
## ---
```

```
## Model selection table
##       (Int)     s1     s2     s3      s6      s9  s1:s2 s1:s3     R21
## 120 -0.7777 -5.918 -2.686 -1.317         -0.8983 -1.929 4.367 0.76650
## 128 -0.7752 -6.096 -2.710 -1.330  0.3021 -0.8771 -1.908 4.357 0.76540
## 104 -0.7703 -5.942 -2.649 -1.688                 -1.917 4.288 0.75810
## 112 -0.7673 -6.169 -2.681 -1.694  0.3858         -1.891 4.277 0.75750
## 88  -0.6208 -4.843 -2.754 -1.291         -0.8620        4.418 0.70080
## 96  -0.6192 -5.191 -2.797 -1.317  0.5538 -0.8238        4.400 0.70150
## 72  -0.6146 -4.873 -2.717 -1.648                        4.342 0.69370
## 80  -0.6132 -5.267 -2.768 -1.659  0.6303                4.324 0.69510
## 70  -0.6285 -4.643        -2.026                        4.515 0.58490
## 86  -0.6338 -4.616        -1.732         -0.7231        4.582 0.58890
## 78  -0.6280 -4.813        -2.034  0.2740                4.509 0.58300
## 94  -0.6333 -4.743        -1.744  0.2050 -0.7081        4.576 0.58670
## 40  -0.4360 -5.310 -2.920 -1.242                 -2.019       0.48260
## 48  -0.4328 -5.646 -2.966 -1.253  0.5677         -1.980       0.48290
## 56  -0.4366 -5.292 -2.940 -1.057         -0.4376 -2.026       0.48240
## 52  -0.4357 -5.344 -3.078                -0.8770 -2.015       0.47120
## 36  -0.4340 -5.410 -3.086                        -1.994       0.46210
## 64  -0.4335 -5.608 -2.982 -1.082  0.5309 -0.4021 -1.989       0.48220
## 60  -0.4330 -5.620 -3.117         0.4619 -0.8552 -1.983       0.47050
## 44  -0.4309 -5.724 -3.130         0.5290         -1.958       0.46210
## 8   -0.2675 -4.174 -2.995 -1.194                              0.41310
## 16  -0.2675 -4.695 -3.060 -1.211  0.8260                      0.41690
## 4   -0.2675 -4.284 -3.155                                     0.39440
## 24  -0.2675 -4.155 -3.014 -1.028         -0.3937              0.41250
## 20  -0.2675 -4.211 -3.148                -0.8211              0.40210
## 32  -0.2675 -4.659 -3.074 -1.066  0.7956 -0.3417              0.41570
## 12  -0.2675 -4.781 -3.218         0.7858                      0.39760
## 28  -0.2675 -4.674 -3.207         0.7269 -0.7881              0.40440
## 6   -0.2675 -3.889        -1.593                             0.28220
## 14  -0.2675 -4.163        -1.607  0.4395                     0.28190
## 22  -0.2675 -3.877        -1.501         -0.2222             0.28070
## 2   -0.2675 -4.018                                           0.24580
## 30  -0.2675 -4.141        -1.526  0.4214 -0.1928             0.28020
## 18  -0.2675 -3.944                       -0.8479             0.25470
## 10  -0.2675 -4.242                0.3583                     0.24510
## 26  -0.2675 -4.131                0.2976 -0.8346             0.25360
## 15  -0.2675        -2.556 -1.415 -2.1090                     0.21990
## 27  -0.2675        -2.730        -2.2070 -1.1670             0.21090
## 31  -0.2675        -2.593 -1.115 -2.1250 -0.6984             0.22410
## 11  -0.2675        -2.731        -2.2190                     0.19200
## 7   -0.2675        -2.595 -1.582                             0.15480
## 23  -0.2675        -2.628 -1.308         -0.6392             0.15850
## 19  -0.2675        -2.792                -1.1900             0.13900
## 3   -0.2675        -2.793                                    0.11860
## 13  -0.2675               -1.732 -2.1550                     0.12430
## 29  -0.2675               -1.505 -2.1670 -0.5366             0.12680
## 25  -0.2675                      -2.2830 -1.1690             0.09995
## 9   -0.2675                      -2.2950                     0.08009
## 5   -0.2675               -1.907                             0.05528
## 21  -0.2675               -1.708         -0.4739             0.05753
## 17  -0.2675                              -1.1930             0.02161
## 1   -0.2675                                                  0.00000
```

```
##         R22 df   logLik   AIC  delta weight
## 120 0.76650  8 -275.108 566.2   0.00  0.529
## 128 0.76540  9 -274.922 567.8   1.63  0.234
## 104 0.75810  7 -277.312 568.6   2.41  0.159
## 112 0.75750  8 -277.019 570.0   3.82  0.078
## 88  0.70080  7 -288.076 590.2  23.94  0.000
## 96  0.70150  8 -287.586 591.2  24.96  0.000
## 72  0.69370  6 -289.652 591.3  25.09  0.000
## 80  0.69510  7 -289.032 592.1  25.85  0.000
## 70  0.58490  5 -305.380 620.8  54.54  0.000
## 86  0.58890  6 -304.575 621.2  54.93  0.000
## 78  0.58300  6 -305.295 622.6  56.37  0.000
## 94  0.58670  7 -304.527 623.1  56.84  0.000
## 40  0.48260  6 -316.292 644.6  78.37  0.000
## 48  0.48290  7 -316.001 646.0  79.79  0.000
## 56  0.48240  7 -316.055 646.1  79.89  0.000
## 52  0.47120  6 -317.403 646.8  80.59  0.000
## 36  0.46210  5 -318.519 647.0  80.82  0.000
## 64  0.48220  8 -315.800 647.6  81.38  0.000
## 60  0.47050  7 -317.215 648.4  82.21  0.000
## 44  0.46210  6 -318.277 648.6  82.34  0.000
## 8   0.41310  5 -322.954 655.9  89.69  0.000
## 16  0.41690  6 -322.408 656.8  90.60  0.000
## 4   0.39440  4 -324.764 657.5  91.31  0.000
## 24  0.41250  6 -322.786 657.6  91.36  0.000
## 20  0.40210  5 -323.902 657.8  91.59  0.000
## 32  0.41570  7 -322.281 658.6  92.34  0.000
## 12  0.39760  5 -324.287 658.6  92.36  0.000
## 28  0.40440  6 -323.488 659.0  92.76  0.000
## 6   0.28220  4 -333.375 674.8 108.53  0.000
## 14  0.28190  5 -333.249 676.5 110.28  0.000
## 22  0.28070  5 -333.332 676.7 110.45  0.000
## 2   0.24580  3 -336.015 678.0 111.81  0.000
## 30  0.28020  6 -333.216 678.4 112.22  0.000
## 18  0.25470  4 -335.282 678.6 112.35  0.000
## 10  0.24510  4 -335.935 679.9 113.65  0.000
## 26  0.25360  5 -335.227 680.5 114.24  0.000
## 15  0.21990  5 -337.485 685.0 118.75  0.000
## 27  0.21090  5 -338.073 686.1 119.93  0.000
## 31  0.22410  6 -337.087 686.2 119.96  0.000
## 11  0.19200  4 -339.387 686.8 120.56  0.000
## 7   0.15480  4 -341.676 691.4 125.14  0.000
## 23  0.15850  5 -341.370 692.7 126.52  0.000
## 19  0.13900  4 -342.620 693.2 127.02  0.000
## 3   0.11860  3 -343.869 693.7 127.52  0.000
## 13  0.12430  4 -343.478 695.0 128.74  0.000
## 29  0.12680  5 -343.270 696.5 130.32  0.000
## 25  0.09995  4 -344.876 697.8 131.54  0.000
## 9   0.08009  3 -346.030 698.1 131.84  0.000
## 5   0.05528  3 -347.373 700.7 134.53  0.000
## 21  0.05753  4 -347.223 702.4 136.23  0.000
## 17  0.02161  3 -349.141 704.3 138.07  0.000
## 1   0.00000  2 -350.245 704.5 138.27  0.000
## Models ranked by AIC(x)
```

El siguiente paso usaremos la función `get.models()` para seleccionar los mejores modelos en función de su AIC o de su "weight" (en caso de que no tengamos ningún modelo con un weight $> 0{,}90$). Así, podemos seleccionar aquellos modelos con una diferencia en el AIC (delta o $\Delta$AIC) igual o inferior a 2, 4 o 6 unidades respecto al modelo con el AIC menor, o seleccionar el mínimo número de mejores modelos cuyo weight conjunto sea $\geq 0{,}95$. De manera alternativa, también nos podríamos quedar con el modelo de mayor sentido ecológico y sea menos complecjo dentro de los que tengan un AIC más bajo ($\Delta\text{AIC} \geq 2$). A día de hoy, no existe consenso sobre qué método seguir (Grueber et al., 2011), así que dependerá de la pregunta ecológica a responder y de la inspección de los tamaños de efecto realizada con anterioridad (Banner and Higgs, 2017).

```
# ejemplo1: subconjunto delta AIC mayor o igual que 2
mod_set <- get.models (mod_d, subset=delta<=2)
# ejemplo2: subconjunto AIC mayor o igual que 0.95
mod_set <- get.models (mod_d, subset = cumsum   (mod_d$weight)<=.95)
```

Finalmente, a través de la función `model.avg()`, procederemos a obtener los coeficientes (tamaño del efecto) promedio y el p valor para cada una de las variables explicativas contenidas en los modelos seleccionados. Esta función nos ofrece dos tipos de promediado que, en ambos casos, proporcionan de una media de los coeficientes ponderada por el weight de cada modelo. El *zero-method average* (*full average* en R) considera el valor de los coeficientes de cada variable explicativa que aparecen en los modelos seleccionados y el valor cero para aquellos modelos que no contengan dicha variable. Por el contrario, el *natural average* (*conditional average* en R) solo promedia los valores de los coeficientes incluidos en los modelos seleccionados. Aunque no existe un consenso sobre cuál de los dos métodos es más apropiado (Grueber et al., 2011), el primero es más conservador. Por lo tanto, la elección de uno u otro dependerá de la pregunta a resolver.

```
mod_av <- model.avg (mod_set, revised.var = TRUE) # model   averaging
summary (mod_av) # resultados del multi-model averaging
```

```
##
## Call:
## model.avg(object = mod_set, revised.var = TRUE)
##
## Component model call:
## lm(formula = y ~ <3 unique rhs>, data = dat)
##
## Component models:
##         df  logLik    AIC delta weight
## 123567   8 -275.11 566.22  0.00   0.57
## 1234567  9 -274.92 567.84  1.63   0.25
## 12367    7 -277.31 568.62  2.41   0.17
##
## Term codes:
##    s1    s2    s3    s6    s9 s1:s2 s1:s3
##     1     2     3     4     5     6     7
##
## Model-averaged coefficients:
## (full average)
##             Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept) -0.77577    0.39688     0.40212   1.929  0.05370 .
## s1          -5.96716    0.48534     0.49160  12.138  < 2e-16 ***
## s2          -2.68593    0.40380     0.40912   6.565  < 2e-16 ***
## s3          -1.38390    0.45772     0.46321   2.988  0.00281 **
## s9          -0.73833    0.52297     0.52703   1.401  0.16124
## s1:s2       -1.92136    0.36946     0.37434   5.133    3e-07 ***
## s1:s3        4.35092    0.40448     0.40979  10.617  < 2e-16 ***
## s6           0.07682    0.29121     0.29430   0.261  0.79408
```

```
##
## (conditional average)
##             Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  -0.7758     0.3969      0.4021   1.929  0.05370 .
## s1           -5.9672     0.4853      0.4916  12.138  < 2e-16 ***
## s2           -2.6859     0.4038      0.4091   6.565  < 2e-16 ***
## s3           -1.3839     0.4577      0.4632   2.988  0.00281 **
## s9           -0.8918     0.4399      0.4457   2.001  0.04539 *
## s1:s2        -1.9214     0.3695      0.3743   5.133    3e-07 ***
## s1:s3         4.3509     0.4045      0.4098  10.617  < 2e-16 ***
## s6            0.3021     0.5152      0.5221   0.579  0.56283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Relative variable importance:
##                      s1   s2   s3   s1:s2 s1:s3 s9   s6
## Importance:          1.00 1.00 1.00 1.00  1.00  0.83 0.25
## N containing models:    3    3    3    3     3     2    1
```

De manera adicional, también podemos promediar las predicciones de los modelos seleccionados, que podremos comparar con las dos estimaciones del model averaging de los coeficientes.

```
# Predicción del modelo que promedia los coeficientes (full)
predict(mod_av, full = T)-> av_pred_full
# Predicción del modelo que promedia los coeficientes (conditional)
predict(mod_av, full = F)-> av_pred_subset
# Predicción del modelo que promedia los valores estiamdos
rowMeans(data.frame(lapply(mod_set, predict))) -> av_pred_res
```

3. **Comprobar y validar las asunciones del modelo**

Actualmente, no existe una manera sencilla de comprobar las asunciones de normalidad de residuos, homocedasticidad y autocorrelación espacial/temporal en los residuos del modelo promedio. Por lo tanto, se pueden comprobar en el modelo global o en cada uno de los modelos seleccionados. Para ver más detalles sobre cómo comprobar las asunciones de los modelos GLMM o GAMM, recomendamos consultar Zuur et al. (2009).

Aunque el *multi-model inference* nos puede ayudar a reducir la incertidumbre en el modelado ecológico, existe un debate activo sobre sus limitaciones reales y las circunstancias en las cuales podemos aplicarla de forma segura (Banner and Higgs, 2017; Tyre, 2017). En cualquier caso, debemos de ser cuidadosos en la selección de variables explicativas y evitar la tentación de usar esta técnica como un sistema automático (no supervisado) de selección de modelos.

# Visualización e interpretación de los resultados

Existen varias formas de interpretar y evaluar la importancia relativa de los efectos aditivos y combinados (interacciones) entre predictores. La mejor manera de interpretar nuestros resultados es comenzar con una visualización de sus efectos. Para ello, podemos construir una gráfica donde se muestre la respuesta de un predictor 1 en función de distintos valores (bajos, intermedios y altos) de un predictor 2. Además, también nos será útil examinar el signo de los coeficientes de los dos estresores individuales y de su interacción. Por conveniencia, llamaremos indicador a la variable respuesta, y estresores 1 y 2 a los dos predictores independientes.

De esta manera, podemos clasificar los efectos en cuatro tipos (Figura 2):

1. *Efectos aditivos.* A nivel ecológico, podemos decir que el efecto de ambos estresores es independiente y aditivo. El efecto final de los dos estresores coincide con la suma de sus efectos individuales. En la

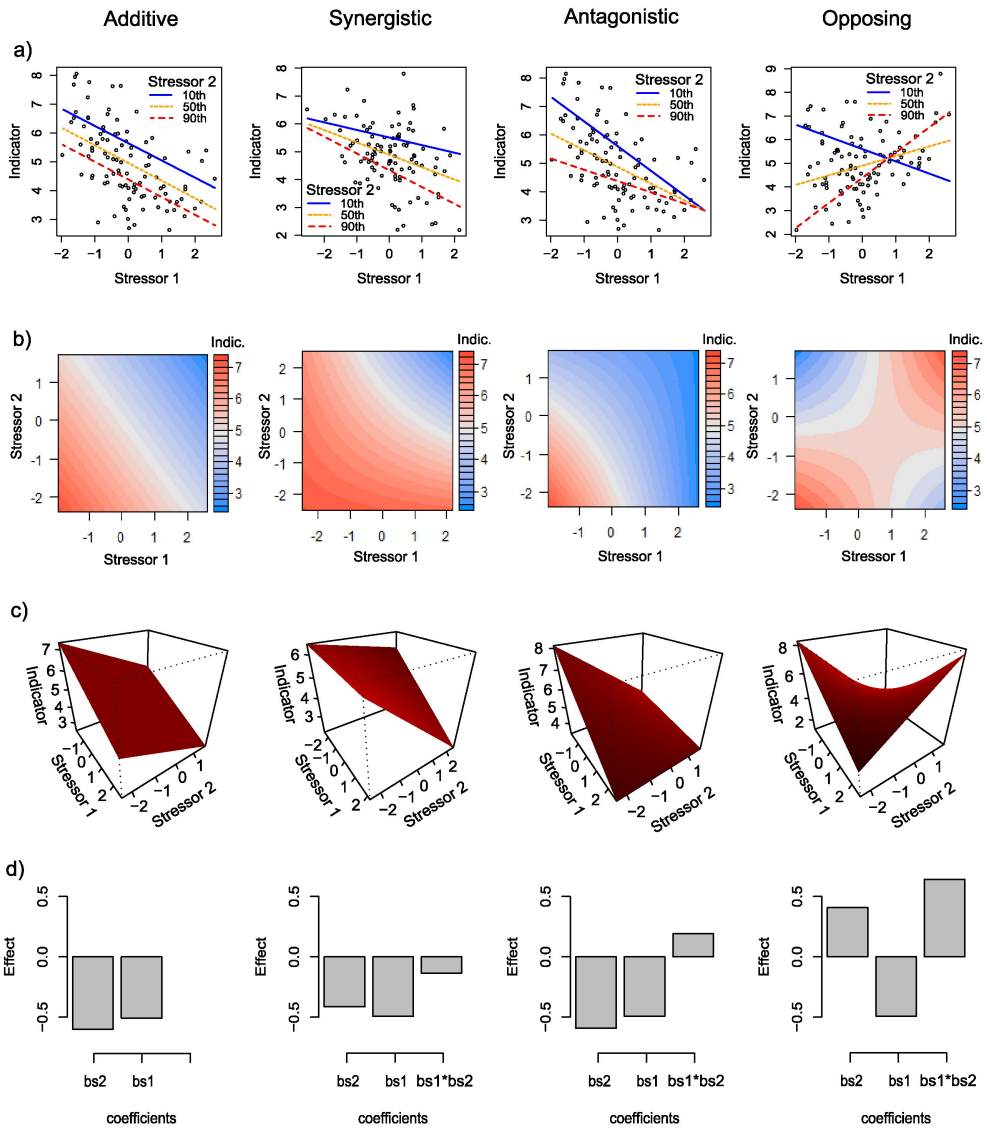



Figure 2: Visualización e interpretación de las interacciones entre predictores (extraída de Feld et al., 2016). En este ejemplo se usan dos estresores como predictores.

visualización observaremos que el indicador (eje y) muestra una respuesta frente al estresor 1 que tienen la misma pendiente para distintos valores del estresor 2 (líneas de respuesta paralelas). Dependiendo de la importancia relativa de ambos estresores, las líneas paralelas estarán más o menos juntas. Los signos de los coeficientes de los estresores pueden ser positivos o negativos, y la interacción estará próxima a cero (normalmente, un p valor alto).

2. *Efectos sinérgicos.* A nivel ecológico, podemos decir que el efecto de ambos estresores es dependiente ya que ambos estresores se potencian y causan un daño mayor al esperado por la suma de sus efectos individuales. En la visualización observaremos que las líneas divergen a medida que el estresor 1 aumenta. En este caso, los coeficientes de los estresores y de la interacción tienen el mismo signo, y normalmente son distintos de cero. En caso de que los dos estresores tengan efectos negativos, veremos como la línea que representa el valor más alto para el estresor 2 muestra una reducción del indicador mucho más intensa que para valores menores. Es importante considerar que esta interacción se puede dar para variables que tienen un efecto positivo (e.g. temperatura y nutrientes para el crecimiento de plantas o algas).

3. *Efectos antagónicos.* A nivel ecológico, interpretaremos que el efecto de ambos estresores es dependiente y son capaces de neutralizarse, causando un efecto combinado menor del esperado por la suma de sus efectos individuales. En la visualización veremos como las líneas convergen cuando los valores del estresor 2 van siendo mayores. En este caso, los coeficientes de los estresores tienen el mismo signo y pero la interacción tiene un signo contrario (normalmente los coeficientes difieren de cero).

4. *Efectos opuestos.* A nivel ecológico se trata de una interacción compleja de interpretar y que nos indica que el estresor 1 puede tener efectos positivos o negativos en el indicador dependiendo de los valores del estresor 2. En la visualización observaremos como las rectas se cruzan cuando los estresores toman valores intermedios. En este caso, los coeficientes de los estresores tienen distinto signo, mientras que la interacción puede tener signo positivo o negativo (normalmente los coeficientes difieren de cero).

En muchas ocasiones uno o varios de los predictores que interaccionan son factores semi-cuantitativos o cualitativos. En estos casos la interpretación de la interacción es más compleja y nos debemos guiar sobre todo por la visualización de las respuestas. Existen más maneras de clasificar e interpretar las interacciones, pero son tema de estudio avanzado (Côté et al., 2016; Feld et al., 2016; Piggott et al., 2015; Schäfer and Piggott, 2018).

Además de esta interpretación, es conveniente hacer una partición de la varianza para cuantificar la importancia relativa de las interacciones respecto a los efectos individuales. La librería `variancePartition` es bastante útil, pero solo funciona con variables cuantitativas y tiene algunas limitaciones respecto al modelo que podemos usar. Solo es posible aplicarla al modelo global o a los modelos individuales, pero no al promedio.

Es recomendable consultar las revisiones y meta-análisis de los efectos de los estresores múltiples disponibles para un amplio rango de varios ecosistemas y tipos de organismo (Cameron et al., 2016; Crain et al., 2008; Jackson et al., 2016; Przeslawski et al., 2015; Velasco et al., 2019).

# Referencias


Banner, K.M., Higgs, M.D., 2017. Considerations for assessing model averaging of regression coefficients: Ecol. Appl. 27, 78-93. doi:10.1002/eap.1419

Barton, K., 2016. MuMIn: Multi-model inference. R package version 1.15.6. Version 1, 18.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5-32.

Burnham, K.P., Anderson, D.R., 2002. Model selection and multimodel inference: A practical information-theoretic approach (2nd ed), Ecological Modelling. doi:10.1016/j.ecolmodel.2003.11.004

Cade, B., 2015. Model averaging and muddled multimodel inferences. Ecology 96, 2370-2382.

Cameron, E.K., Vilà, M., Cabeza, M., 2016. Global meta-analysis of the impacts of terrestrial invertebrate invaders on species, communities and ecosystems. Glob. Ecol. Biogeogr. 25, 596-606. doi:10.1111/geb.12436

Côté, I.M., Darling, E.S., Brown, C.J., 2016. Interactions among ecosystem stressors and their importance in conservation. Proc. R. Soc. B Biol. Sci. 283, 20152592. doi:10.1098/rspb.2015.2592

Crain, C.M., Kroeker, K., Halpern, B.S., 2008. Interactive and cumulative effects of multiple human stressors in marine systems. Ecol. Lett. 11, 1304-1315. doi:10.1111/j.1461-0248.2008.01253.x

Crawley, M.J., 2014. Statistics. An introduction Using R, 2nd ed. Wiley.

Cutler, D.R., Edwards Jr, T.C., Beard, K.H., Cutler, A., Hess, K.T., Gibson, J., Lawler, J.J., 2007. Random Forests for Classification in Ecology 88, 2783-2792. doi:10.1890/07-0539.1

Elith, J., Leathwick, J.R., Hastie, T., 2008. A working guide to boosted regression trees. J. Anim. Ecol. 77, 802-813. doi:10.1111/j.1365-2656.2008.01390.x

Feld, C.K., Segurado, P., Gutiérrez-Cánovas, C., 2016. Analysing the impact of multiple stressors in aquatic biomonitoring data: A 'cookbook' with applications in R. Sci. Total Environ. 573. doi:10.1016/j.scitotenv.2016.06.243

Grueber, C.E., Nakagawa, S., Laws, R.J., Jamieson, I.G., 2011. Multimodel inference in ecology and evolution: Challenges and solutions. J. Evol. Biol. 24, 699-711. doi:10.1111/j.1420-9101.2010.02210.x

Harrell, F.E., 2001. Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis. Springer.

Hijmans R.J., S., P., J., L., J., E., 2016. dismo: Species Distribution Modeling.

Ishwaran, H., Gerds, T.A., Kogalur, U.B., Moore, R.D., Gange, S.J., Lau, B.M., 2014. Random survival forests for competing risks. Biostatistics 15, 757-773. doi:10.1093/biostatistics/kxu010

Jackson, M.C., Loewen, C.J.G., Vinebrooke, R.D., Chimimba, C.T., 2016. Net effects of multiple stressors in freshwater ecosystems: A meta-analysis. Glob. Chang. Biol. 22, 180-189. doi:10.1111/gcb.13028

Johnson, J.B., Omland, K.S., 2004. Model selection in ecology and evolution. Trends Ecol. Evol. 19, 101-108. doi:10.1016/j.tree.2003.10.013

Piggott, J.J., Townsend, C.R., Matthaei, C.D., 2015. Reconceptualizing synergism and antagonism among multiple stressors. Ecol. Evol. 5, 1538-1547. doi:10.1002/ece3.1465

Przeslawski, R., Byrne, M., Mellin, C., 2015. A review and meta-analysis of the effects of multiple abiotic stressors on marine embryos and larvae. Glob. Chang. Biol. 21, 2122-2140. doi:10.1111/gcb.12833

Ridgeway, G., 2015. gbm: Generalized Boosted Regression Models.

Schäfer, R.B., Piggott, J.J., 2018. Advancing understanding and prediction in multiple stressor research through a mechanistic basis for null models. Glob. Chang. Biol. 24, 1817-1826. doi:10.1111/gcb.14073

Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A., 2008. Conditional variable importance for random forests. BMC Bioinformatics. doi:10.1186/1471-2105-9-307

Strobl, C., Malley, J., Gerhard T., 2009. Characteristics of classification and regression trees, bagging and random forests. Psychol. Methods 14, 323-348. doi:10.1037/a0016973.An

Tyre, D., 2017. Does model averaging make sense? [WWW Document]. A few cheap shots. URL http://atyre2.github.io/2017/06/16/rebutting_cade.html

Velasco, J., Gutiérrez-Cánovas, C., Botella-Cruz, M., Sánchez-Fernández, D., Arribas, P., Carbonell, J.A., Millán, A., Pallarés, S., 2019. Effects of salinity changes on aquatic organisms in a multiple stressor context. Phil. Trans. R. Soc. B 374, 20180011. doi:10.1098/RSTB.2018.0011

Walker, J.A., 2018. On model averaging partial regression coefficients. bioRxiv. doi:10.1111/j.1468-5922.2007.00690.x

Zuur, A.F., Ieno, E.N., Walker, N.J., Saveliev, A.A., Smith, G.M., Ebooks Corporation., 2009. Mixed effects models and extensions in ecology with R, Statistics for Biology and Health. doi:10.1007/978-0-387-87458-6

## Lecturas recomendadas

**Classification and regression trees (CART)**

Boosted Regression Trees for ecological modeling (paper)

A very basic introduction to Random Forests using R

Random Survival Forests for R

Las limitaciones de los CART, en Dynamic Ecology por Brian McGill (artículos 1 y 2)

**Data exploration**

A protocol for data exploration to avoid common statistical problems

**GLM, GLMM and GAMMs**

A protocol for conducting and presenting results of regression-type analyses

Three points to consider when choosing a LM or GLM test for count data

GLMM FAQ

Overview GAMM analysis of time series data

Modelling seasonal data with GAMs

Additive modelling (GAMM) global temperature time series: revisited

Climate change and spline interactions (GAMs)

**Model interpretation and p-values**

Ajuste, interpretación y presentación de modelos lineales: el valor p no es suficiente

Assessing hypotheses and simulation-based approaches

Advancing understanding and prediction in multiple stressor research through a mechanistic basis for null models

The ASA's Statement on p-Values: Context, Process, and Purpose

**Multi-model inference criticisms and limitations**

Does model averaging make sense?

Model averaging and muddled multimodel inferences

Considerations for assessing model averaging of regression coefficients