multifit: una función de R

Análisis multi-escala en ecología del paisaje

Pablo Yair Huais

Mayo 2018

Descripción

La función *multifit* fue creada para automatizar el análisis multi-escala en ecología del paisaje. El usuario proporciona un data.frame que contiene una columna con la variable de respuesta estudiada y varias columnas que representan un atributo de paisaje particular en varias escalas espaciales. Además, el usuario debe proporcionar el tipo de modelo que se aplicará en el análisis, junto con una fórmula y cualquier otro argumento relevante (por ejemplo, factores bloque, covariables, efectos aleatorios, etc.), así como el criterio que se utilizará para la selección del mejor modelo. La salida de la función incluye los siguientes elementos: un gráfico que representa la fuerza de cada modelo, un gráfico opcional que muestra los coeficientes estimados del modelo de la variable de respuesta para cada modelo y una lista que contiene información relevante sobre los modelos (incluyendo el/los gráfico/s y los modelos como objetos de R).

Dependencias

La función requiere el paquete correspondiente que contiene a la función que se utilizará para ejecutar los modelos.

Función

```
multifit(mod, multief, formula = NULL, data, args = NULL, criterion = "AIC", site_id = NULL,
         signif = TRUE, alpha = 0.05, print_sum = FALSE, plot_est = FALSE,
         xlab = "Radius [m]", ylab = NULL, labels = NULL, type = "b", pch = c(1, 16))
```

Argumentos

`mod`	cadena de texto que describe el tipo de modelo a ser aplicado (ver detalles)
`multief`	caracter. Un vector conteniendo el nombre de las columnas del data.frame que contienen los valores del atributo del paisaje a distintas escalas espaciales
`formula`	formula a ser aplicada a cada modelo, etiquetando el atributo del paisaje a ser analizado como `multief` (ver detalles)
`data`	data.frame conteniendo a la variable respuesta y a los atributos del paisaje en varias escalas espaciales
`args`	vector de caracteres con cualquier otro argumento adicional de los modelos (ver detalles)
`criterion`	caracter. Criterio de selección del mejor modelo. Hasta ahora, una de tres opciones (`"AIC"`, `"BIC"` o `"R2"`) o una función definida por el usuario especificando el nombre de la función en un primer elemento y el criterio de selección del modelo (`"max"` o `"min"`) en un segundo elemento (ver detalles)
`site_id`	caracter. Una cadena de texto conteniendo el nombre de la columna del data.frame que contiene la identidad de los diferentes sitios. Únicamente relevante para la tabla de resumen el atributo del paisaje (ver detalles)
`signif`	lógico. ¿Diferenciar los modelos no significativos de los significativos en el gráfico con diferentes estilos de puntos?
`alpha`	valor numérico, entre 0 y 1. Nivel de significación estadística (únicamente relevante si `signif = TRUE`, ver detalles)
`print_sum`	lógico. ¿Mostrar el summary del mejor modelo?
`plot_est`	lógico. ¿Graficar los coeficientes estimados de los modelos?
`xlab`	caracter. Un título para el eje x
`ylab`	caracter. Un título para el eje y del gráfico que muestra la fuerza de los modelos
`labels`	vector de caracteres con las etiquetas de los modelos en el eje x
`type`	caracter. ¿Qué tipo de gráfico debiera ser ploteado (ver detalles)?

`pch`	valor numérico con dos elementos, conteniendo los códigos de las estilos de puntos para modelos no significativos y significativos, respectivamente (ver detalles)

Detalles

El objetivo de esta función es automatizar el análisis de variables ecológicas en relación con un atributo de paisaje particular, a través de un enfoque de multi-escala. De esta forma, *multifit* permite al usuario ejecutar muchos modelos estadísticos al mismo tiempo (es decir, un modelo por escala espacial), simplificando el análisis y la selección de la escala espacial apropiada para la variable de respuesta proporcionada.

En primer lugar, el usuario debe poseer un data.frame con al menos una columna con la variable de respuesta a analizar, y varias columnas que contengan los valores de un atributo de paisaje (por ejemplo, cantidad de hábitat o número de parches) a diversas escalas espaciales (es decir una columna por escala espacial). Este data.frame debe especificarse en el argumento `data` de la función.

El usuario debe proporcionar el tipo de modelo estadístico que se aplicará en el análisis, especificándolo como un carácter en el argumento `mod` (por ejemplo,"lm" para un modelo lineal clásico). El usuario debe cargar previamente el paquete correspondiente que contiene la función antes de ejecutar el análisis multi-escala. En el argumento `multief`, el usuario debe proporcionar un vector de caracteres que represente los nombres de las columnas que contienen los valores del atributo de paisaje (por ejemplo,`multief = c ("R_500", "R_1000", "R_1500")`, que se referiría, por ejemplo, al atributo de paisaje a tres escalas espaciales diferentes: radios de 500, 1000 y 1500 m). Es importante proporcionar los elementos del vector en un orden que tenga sentido (que lógicamente sería un orden que represente un aumento en la escala espacial), ya que este orden se utilizará para el análisis multi-escala y para los gráficos.

El argumento `formula` debe llenarse con la fórmula estadística que se aplicará en cada modelo. Esto debe incluir al menos la variable de respuesta principal y una variable predictora llamada '`multief`' (e.g. `formula = response_variable ~ multief`). La función reconocerá esta cadena de texto en particular (i.e. `multief`) en cada modelo como la variable de predicción que contiene el atributo de paisaje en cada escala espacial. Si la definición del modelo de la función especificada en `mod` no incluye un argumento llamado `formula`, entonces las variables de respuesta y predicción se pueden definir en el argumento `args` (ver a continuación) en la forma que requiera la definición del modelo.

El usuario puede agregar tantos argumentos como sea necesario para ejecutar los modelos en cada escala espacial. Estos deben agregarse en el argumento `args` como un vector de caracteres, cada elemento representando un argumento particular escrito como el usuario lo haría en un análisis individual (por ejemplo, suponiendo un modelo lineal clásico,`args = c ("na.action = na.omit","singular.ok = FALSE")`). Como se explicó anteriormente, `args` puede incluir las variables de respuesta y predictiva (i.e.`multief`) para aquellas funciones que no incluyen un argumento llamado `formula` (por ejemplo, la función 'lme' del paquete *nlme*, Pinheiro et al. 2018) al especificarlos en los argumentos correspondientes.

El argumento `criterion` debe incluir el criterio que se utilizará para la selección del mejor modelo entre las diversas escalas espaciales (es decir, el que tiene la relación más fuerte con la variable respuesta). Hasta ahora, *multifit* permite elegir entre tres opciones: `"R2"` (para R-cuadrado, es decir, coeficiente de determinación), `"AIC"` (para el Criterio de Información de Akaike), y `"BIC"` (para el Criterio de Información Bayesiano). El usuario debe tener en cuenta si el tipo de modelo definido en `mod` permite el cálculo del criterio especificado. Si no fuese así, *multifit* recomendará el uso de otro. La función también ofrece la posibilidad de especificar una función definida por el usuario para el cálculo de un criterio diferente. Esta función debe tener la capacidad de calcular un valor de criterio particular a partir de un objeto que contiene los modelos estadísticos en cada escala espacial (es decir, la salida de un modelo cuando se ejecuta individualmente). Si este es el caso, el usuario debe especificar el nombre de la función en un primer elemento del vector, y el criterio de selección del modelo (`"max"` o `"min"` del valor del criterio) en una segundo elemento (e.g. `criterion = c ("mi_funcion","max")`).

El argumento `site_id` debe incluir el nombre de la columna en el data.frame que contiene la identidad de los sitios en los cuales se recopiló la variable de respuesta. Aclarar esto solo es necesario para calcular correctamente el n, la media y la mediana del atributo de paisaje en cada escala espacial (que se incluyen como una tabla de resumen en la salida de la función). En caso de colocar `NULL`, el n, la media y la mediana del atributo de paisaje no se calcularán, debido a las posibles ambigüedades en los valores del atributo de paisaje entre sitios diferentes.

El argumento `signif` pregunta si los puntos de la gráfica deben dibujarse diferencialmente de acuerdo con la significación estadística de los coeficientes estimados del modelo, mientras que el argumento `alpha` define el nivel estadístico. El argumento `plot_est` pregunta si un diagrama que represente los coeficientes estimados del modelo en cada escala

espacial debe mostrarse en un gráfico, aparte de la gráfica predeterminada. El argumento `print_sum` pregunta si el resumen (summary) del modelo seleccionado (es decir, el mejor modelo según el criterio definido) debe imprimirse en la consola.

El usuario puede cambiar algunas características estéticas de los gráficos, como `xlab` (las etiquetas del eje x) o `type` (el tipo de gráfico, busque ?type). El argumento `ylab` cambia el título del eje y de la gráfica que muestra la fuerza de los modelos en cada escala espacial. El argumento `pch` define la forma de los puntos que se trazarán, en un vector numérico de dos elementos, para los modelos no significativos y significativos respectivamente (solo relevante si `signif = TRUE`). De forma predeterminada, la función representará los modelos no significativos como puntos con relleno blanco y los significativos como puntos con relleno negro. El argumento `labels` permite al usuario especificar un vector de caracteres con nombres para cada modelo en el eje x, que debe tener la misma cantidad de elementos que el número de escalas espaciales analizadas.

Salida de la función

multifit devuelve una lista con los siguientes componentes:

`lands_summary`	un data.frame conteniendo una tabla de resumen de los valores del atributo del paisaje en cada escala espacial: n, min, max, rango, media y mediana
`summary`	un data.frame conteniendo información relevante de los modelos, incluyendo el valor del criterio, los coeficientes estimados de los modelos y los valores p
`plot`	un gráfico que muestra la fuerza de los modelos en cada escala espacial de acuerdo al criterio especificado, junto a un gráfico opcional que plotea los coeficientes estimados de los modelos
`models`	una lista conteniendo los modelos como objetos de R de todas las escalas espaciales. Estos serán útiles para análisis *a posteriori* de modelos particulares
`warnings`	una lista conteniendo las advertencias, si ocurrieron, durante el análisis de los modelos de cada escala espacial
`messages`	una lista conteniendo los mensajes, si ocurrieron, durante el análisis de los modelos de cada escala espacial

La función muestra un gráfico, que es el mismo que se incluye en el componente 'plot' de la lista devuelta (disponible como un objeto R para que el usuario lo muestre en el dispositivo gráfico de R en cualquier otro momento).

Nota

Por ahora, *multifit* se puede aplicar para efectos univariados únicamente (es decir, con un único atributo de paisaje). Por otro lado, *multifit* ha sido probado con funciones de modelos de los paquetes *stats* ("lm" y "glm"; R Core Team 2017), *lme4* ("lmer" y "glmer"; Bates et al. 2015), *glmmadmb* ("glmmADMB"; Fournier et al. 2012), *glmmTMB* ("glmmTMB"; Magnusson et al. 2017), * nlme * ("lme", Pinheiro et al. 2018) y *pscl* ("zeroinfl" y "hurdle"; Zeileis et al. 2008). Sin embargo, la función debería funcionar con otros modelos y paquetes, y se actualizará en el futuro.

Autor

Pablo Yair Huais – phuais@gmail.com

Ejemplo

```
#           #
# Ejemplo 1 #
#           #

# Leer data.frame: datos que simulan la riqueza de aves a lo largo de un
# gradiente de cantidad de bosque.
#  Esta comtiene una columna con la variable respuesta, en este caso la riqueza de
#  aves, y 10 columnas conteniendo los valores de cantidad de bosque a 10 escalas espaciales
```

```r
#  (desde 500 hasta 5000, cada 500 m). Además, otra columna llamada 'site' contiene
#  la identidad de los sitios en donde la variable respuesta se hubiese obtenido.
#  Hay en total 50 sitios, 10 medidas de riqueza de aves por sitio, lo que hace un total de
#  500 medidas (i.e. filas en la tabla).
# Puede encontrar y descargar este data.frame en el siguiente link:
#  https://github.com/phuais/multifit/blob/master/fake_data/bird_richness.csv

data <- read.csv("fake_data/bird_richness.csv")

# Crear un objeto con multifit
#  En este caso, se relaciona la variable respuesta 'S' (riqueza) con la proporción de
#  bosque en las escalas espaciales especificadas. Se aplica un GLMM con la función 'glmer'
#  del paquete 'lme4' (Bates et al. 2015) y el criterio de AIC para el proceso de selección
#  de modelos (la opción predeterminada). Nótese que el efecto a cada escala espacial es
#  especificado como los nombres de las columnas correspondientes en data. También nótese
#  que se define un efecto aleatorio 'site' para el GLMM.

library(lme4)
fits <- multifit(mod = "glmer", multief = colnames(data)[3:12],
        formula =  S ~ multief + (1|site), args = c("family = poisson"),
        data = data, criterion = "AIC", plot_est = T)

# Una vez que los modelos fueron ejecutados, un gráfico se mostrará y el objeto se puede explorar
# Muestra una tabla resumen con los valores del atribute del paisaje en cada escala espacial
fits$lands_summary

# Muestra una tabla resumen con información relevante para el análisis multi-escala
fits$summary

# Muestra el gráfico nuevamente en el dispositivo gráfico de R
fits$plot

# Muestra un modelo particular como objeto de R
fits$models$R_2500

# Y su 'summary'
summary(fits$models$R_2500)

# Chequea por posibles advertencias y mensajes de un modelo en particular
fits$warnings$R_2500
fits$messages$R_2500

#           #
# Ejemplo 2 #
#           #

# Puede definirse un criterio propio de selección de modelos con una función
#  definida por el usuario, por ejemplo, un pseudo-R-cuadrado para GLMMs,
#  el cual puede calcularse con la función r.squaredGLMM del paquete MuMIn (Bartoń 2018):

library(MuMIn)
pseudo_R <- function(x){
  as.numeric(suppressMessages(r.squaredGLMM(x)[1]))
}

# Y luego ejecutar multifit definiendo la función y el criterio de selección
#  en el argumento criterion. En este caso, como estoy definiendo un criterio
```

```
#  de tipo R-cuadrado, el mejor modelo será aquel que tenga el valor máximo de
#  este criterio. Por ello, se especifica "max" como segundo elemento del argumento:

fits <- multifit(mod = "glmer", multief = colnames(data)[3:12],
                 formula =  S ~ multief + (1|site), args = c("family = poisson"),
                 data = data, criterion = c("pseudo_R", "max"), plot_est = T)
```

Referencias

Bates D., Maechler M., Bolker B., Walker S. (2015). Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48.

Bartoń K. (2018). MuMIn: Multi-Model Inference. R package version 1.40.4.

Fournier D.A., Skaug H.J., Ancheta J., Ianelli J., Magnusson A., Maunder M., Nielsen A., Sibert J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. Optimization Methods and Software, 27(2), 233-249.

Magnusson A., Skaug H.J., Nielsen A., Berg C.W., Kristensen K., Maechler M., van Bentham K. J., Bolker B.M., Brooks M.E. (2017). glmmTMB: Generalized Linear Mixed Models using Template Model Builder. R package version 0.1.3.

Pinheiro J., Bates D., DebRoy S., Sarkar D., R Core Team (2018). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-131.1.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Zeileis A., Kleiber C., Jackman S. (2008). Regression Models for Count Data in R. Journal of Statistical Software 27(8).