Revista Ciencias del Mar, UAS

Enero - Marzo 2024 Núm. 2 Vol.1

UNIVERSIDAD AUTÓNOMA DE SINALOA

E-ISSN (en trámite)

Revista CIMAR, UAS

REVISTA DE LA FACULTAD DE CIENCIAS DEL MAR E-ISSN (en trámite)

Nota Científica

Uso de R para ajustar un modelo de crecimiento individual como en Microsoft Excel

Using R to fit an individual growth model similar to Microsoft Excel

1. Jorge Payan-Alejo

0000-0003-4636-0274

Facultad de Ciencias del Mar, Universidad Autónoma de Sinaloa, Paseo Claussen S/N, 82000, Mazatlán, Sinaloa, México.
Autor de correspondencia: jorge.payan.facimar@uas.edu.mx

2. Juan Roberto Felipe Vallarta-Zárate

0000-0001-8654-8728

Dirección de Investigación Pesquera en el Atlántico, Instituto Mexicano de Investigación en Pesca y Acuacultura Sustentable. Av. México #190 Col. Del Carmen. CP 04100, Coyoacán, CDMX.

3. Jesús Ramón Rendón-Martínez

0009-0002-7954-8716

Instituto Mexicano de Investigación en Pesca y Acuicultura Sustentable
Av. México #190 Col. Del Carmen. CP 04100, Coyoacán, CDMX.

4. Efrain Delgado-Robles

0000-0002-4799-0836

Posgrado en Ciencias en Recursos Acuáticos, Facultad de Ciencias del Mar, Universidad Autónoma de Sinaloa, Paseo Claussen S/N C.P. 82000, Mazatlán, Sinaloa, México.

Uso de R para ajustar un modelo de crecimiento individual como en Microsoft Excel.

Using R to fit an individual growth model similar to Microsoft Excel

RESUMEN

El lenguaje de programación R, distribuido como software de uso libre (Licencia Pública General de GNU), ha sido ampliamente usado para el análisis de datos en muchas áreas, incluidas aquellas relativas a las ciencias pesqueras. En este manuscrito se propone y prueba un procedimiento mediante algunas funciones de código abierto en R, para optimizar los parámetros del modelo de crecimiento individual de von Bertalanffy, ajustado a una estructura de error aditivo y multiplicativo por el método de verosimilitud. La función mostró buen desempeño al coincidir en la estimación de los parámetros optimizados con datos de repositorios virtuales, por lo que se recomienda su uso.

***Palabras clave**:* Código, ajuste, parámetros, logverosimilitud, incertidumbre.

ABSTRACT

R programming language, distributed as a free use software (General Public License, GNU), has been widely used for data analysis in many disciplines, included those related to Fisheries Sciences. In this manuscript, it is proposed and proved a procedure using some open-source code functions in R, in order to optimize the parameters of von Bertalanffy individual growth model, which was fitted to additive and multiplicative error structure by likelihood method. The function showed good performance obtaining coincidences with the optimized parameters of the virtual repositories data, consequently, its use highly recommended.

Key words: Code, fit, parameters, log likelihood, uncertainty.

INTRODUCCIÓN

El lenguaje de programación R (Posit Team, 2023) es un software de uso libre para el desarrollo de computación estadística y gráfica, en años recientes se ha incrementado su utilización para el análisis de datos en diferentes campos del conocimiento, entre las cuales se encuentra el área de las ciencias pesqueras (Ogle, 2016; Haddon, 2021).

Existen libros de ciencia pesquera con ejercicios prácticos en plataformas computacionales que se utilizan comúnmente en ambientes como Microsoft Excel (Haddon, 2001, 2011); sin embargo, se han realizado avances enfocados en la migración de análisis y procesos hacia lenguajes innovadores como R. Un ejemplo de esto se puede encontrar en los ejercicios de biología pesquera de Haddon (2011), los cuales fueron actualizados y programados en el ambiente de R (Haddon, 2021). Podría resultar complicado para biólogos pesqueros en formación, con menos experiencia en programación, el hecho de adaptar códigos prediseñados en R para realizar análisis estadísticos y generación de gráficos con sus propios datos.

El desarrollo de ambientes de programación como R, ofrece la posibilidad de realizar análisis estadísticos complejos que antes requerían potentes computadoras y procesos tan largos que incluso

duraban varios días. El manejo de R permite realizar múltiples tratamientos a los datos de interés mediante una interfaz intuitiva y amigable con el usuario como R studio (Posit Team, 2023).

En este documento se utilizará R con R studio, demostrando a los usuarios una forma amigable de interacción, en donde se realizará un ejercicio de ajuste del modelo de crecimiento individual de von Bertalanffy (von Bertalanffy, 1938) y remuestreo de datos para estimar los incertidumbre de los parámetros del modelo.

MATERIAL Y MÉTODOS

En este ejercicio se utilizan los datos edad-talla de: Pacific hake *Merluccius productus* (Kimura, 1980), Pez vela *Istiophorus platypterus* (Cerdenares-Ladrón De Guevara, Morales-Bojórquez, Rodríguez-Sánchez, 2011), Blackdrum *Pogonias cromis* (Ogle, 2016), Redfish, *Centroberyx affinis* (Haddon, 2021) y pinfish *Lagodon rhomboides* (Nelson, 2023).

Estos datos se pueden capturar en un archivo Excel con el formato denominado *delimitado por comas* (.csv) y guardar en el directorio de trabajo.

En el siguiente paso debemos indicarle a R que realice la lectura del archivo que acabamos de crear. En resumen, la lectura de los datos en formato csv se realiza mediante el comando *read.csv*, por ejemplo:

```
datos = read.csv(./"datos_edad_talla.csv", header = TRUE) # Forma de cargar datos.
```

Nótese que mediante los símbolos **./** le estamos indicando a R que busque el archivo en la carpeta raíz donde se encuentra el archivo csv.

Ahora los datos de edad y talla se encuentran en una variable que creamos de nombre *datos* y para llamar la información se crean variables edad = datos$edad y talla = datos$talla. Es importante notar que estamos creando un objeto de nombre *datos* y mediante los símbolos = estamos asignando a nuestro objeto los datos. Pero se pueden crear variables como "*un vector*", con la información de entrada, que se grafican y se utilizan para ajustar el modelo de crecimiento individual. A continuación, para cada conjunto de datos, se nombra la variable edad y

talla seguida de las iniciales que indican pertenencia o identidad por ejemplo:

Datos de Kimura (1980):

```
edad = c(1,2,3.3,4.3,5.3,6.3,7.3,8.3,9.3,10.3,11.3,12.3,13.3) # la letra h indica hembras
talla = c(15.4,28.03,41.18,46.2,48.23,50.26,51.82,54.27,56.93,58.93,59,60.91,61.83)
edad = c(1,2,3.3,4.3,5.3,6.3,7.3,8.3,9.3,10.3,11.3) # la letra m indica machos
talla = c(15.4,26.93,42.23,44.59,47.63,49.67,50.87,52.3,54.77,56.43,55.88)
Datos de Cerdenares-Ladrón De Guevara, Morales-Bojórquez, Rodríguez-Sánchez (2011).
edad = c(1:11) # la letra pv indica pez vela
talla = c(65.1,102.8,125.4,142.4,151.8,161.3,164.8,168.7,172.8,173.8,187.5)
Datos de Ogle (2016).
library(FSAdata) # Cargar el paquete para usar los datos edad-talla
datos = subset(BlackDrum2001,BlackDrum2001$otoage<50 &
BlackDrum2001$sex=="male") # filtro para replicar el ejercicio de Ogle (2016).
datos = datos[order(datos$otoage),] # ordena los datos de edad de menor a mayor.
edad = datos[,9] # el [,9] indica la columna para los datos edad en el archivo virtual
talla = datos[,7] # el [,7] indica la columna para los datos talla en el archivo virtual
Datos de Haddon (2021).
library(MQMF) # cargar el paquete para usar los datos edad-talla
edad = LatA[,1] # el [,1] indica la columna para los datos edad en el archivo virtual
talla = LatA[,2] # el [,2] indica la columna para los datos talla en el archivo virtual
Nelson (2023).
library(fishmethods) # cargar el paquete para usar los datos edad-talla
datos = pinfish[order(pinfish$age),] # ordena los datos de edad de menor a mayor.
edad = datos[,3] # el [,3] indica la columna para los datos edad en el archivo virtual
talla = datos[,2] # el [,2] indica la columna para los datos talla en el archivo virtual
A continuación se muestra el código para graficar los datos edad-talla, incluyendo algunas
modificaciones de formato (la figura es el ejemplo de Haddon, 2021).
plot(edad, talla, bty = "L", # bty es edición de las líneas del marco
xlab="Edad (años)", ylab="Longitud (cm)", pch=1, # pch es la forma del símbolo
main="von Bertalanffy Aditivo", cex=1.2, # cex es el tamaño del símbolo
cex.lab=1.2, cex.axis=1.2, # cex.lab y cex.axis edita etiquetas y números
xlim=c(0, max(edad)), ylim = c(0, max(talla))) # xlim y ylim escalas en ambos ejes
```

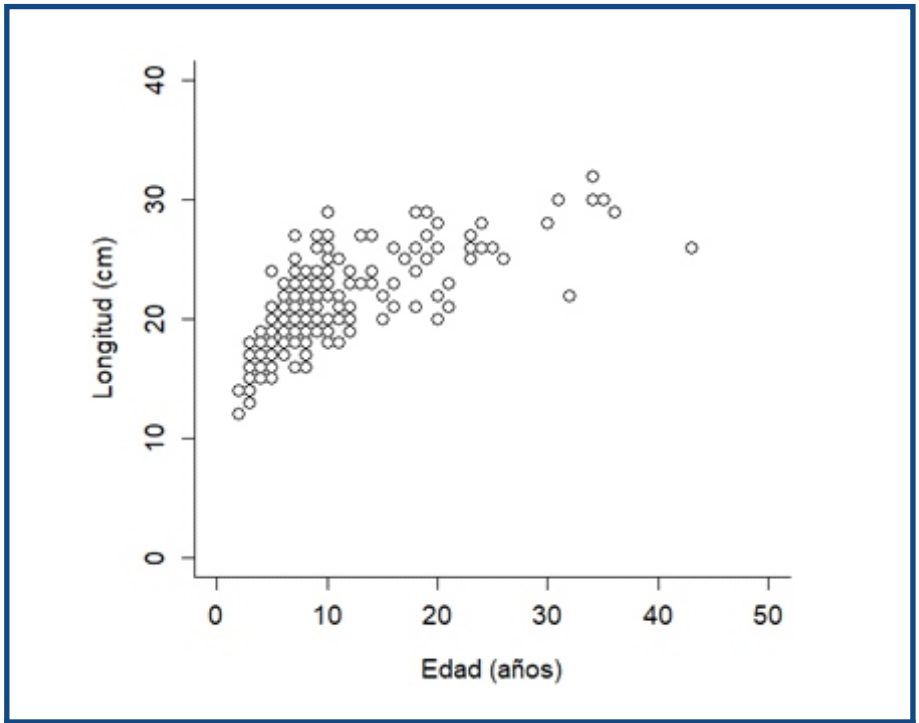

Figura 1. Dispersión de los datos edad-talla Redfish *Centroberyx affinis* (Haddon, 2021).

En R podemos crear funciones que nos permiten ejecutar procesos iterativos, estas funciones pueden ser empleadas para realizar ese proceso en diferentes conjuntos de datos. Crearemos una función a la cual nombraremos *función madre o base*, con ella se puede crear cualquier otro modelo. La función madre se construyó para estimar los parámetros del modelo de crecimiento por el *método de logverosimilitud.*

La función se construye con el modelo de crecimiento individual de von Bertalanffy (1938).

$$L_t = L_\infty\left(1 - e^{-(k*(t-t_0))}\right)$$

Donde: L_t es la longitud L a la edad t, L_∞ es la longitud máxima promedio (asintótica) de la población, k es tasa de crecimiento, t es la edad y t_0 es la edad teórica a la cual la talla es cero.

En los datos edad-talla se explora ajustar el modelo con error aditivo (los errores tienen distribución normal), donde la desviación estándar se estima como:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \left(L_i - \hat{L}_i\right)^2}{n}}$$

Donde σ es la desviación de los datos, *Li* es igual a la longitud observada y *L î* es la longitud estimada y *n* el tamaño de la muestra

A continuación se muestra el código para ajustar el modelo con la función logverosimilitud asumiendo *error aditivo*. En cada línea hay comentarios después del símbolo #, para indicar que se hace dentro de la función. Antes de iniciar es preciso integrar en dos vectores los datos correspondientes a la edad y a la talla.

```
logvero = function(pars) { # inicia creación de la función
Linf = pars[1] # asignación de L infinita
k = pars[2] # asignación de k
t0 = pars[3] # asignación del t0
sigma = pars[4] # asignación de la desviación estándar
n = length(edad) # número de observaciones
lv = 0 # es el punto de inicio de logverosimilitud
for(i in 1:n) { # inicia el loop para los n datos observados
    modelo = (Linf*(1-exp(-k*(edad[i]-t0)))) # estructura del modelo analizado
    lv = lv + log(dnorm((talla[i]), (modelo), sigma)) # suma de logverosimilitud
}
-lv # se invierte la logverosimilitud, porque en R no se puede maximizar
} # Termina la estructura de la función
```

Posteriormente la optimización de los parámetros se realiza con la función predeterminada *nlminb()*, en este caso asignada a un objeto llamado resMA, donde primero se ingresa de la información de los parámetros a cambiar (*siguiendo el orden asignados en la función madre*) y después la función objetivo (logvero).

Considere que los valores iniciales de los parámetros se pueden obtener de un estudio previo, pero cuando no se tiene esa información, el valor de longitud asintótica (L_∞) se considera a la talla máxima de los datos, en la tasa de crecimiento (k) se asigna un medio cuando los datos tienen edad igual o menor a 10 años y un tercio cuando es mayor a 10 años (el código se encuentra condicionado para realizar el procedimiento), esto considerando que k varía de valores mayor a cero e igual o menor a uno, en el valor de la edad teórica (t_0) se asignó al negativo de la unidad, dado que generalmente este valor es menor a cero y el negativo de dos, finalmente un *proxi* de sigma (σ) es la desviación estándar de los datos de edad o talla. Los resultados de interés de resMA se encuentran en $par (parámetros) y $objetive (logverosimilitud).

La estructura de *nlminb()* se asigna a una variable nombrada resMA (resultado Modelo Aditivo) y resMM (resultado Modelo Multiplicativo) para diferenciar los resultados entre estructura de error, pero cuando solo es un modelo con la estructura de nlminb es suficiente.

```
resMA = suppressWarnings(nlminb(start = c(max(talla), ifelse(max(edad) >= 10, 0.3, 0.5),-1, sd(talla)), logvero)) # suppressWarnings es para evitar caer en error.
```

El proceso de optimización se visualiza con *resMA*, los resultados de interés se encuentran en $par (parámetros) y $objetive (logverosimilitud), esto se debe a que en la función se indica que se estiman parámetros y logvero es la función objetivo, los cuales tienen el orden asignado en la estructura de la función.

```
resMA # para ver el resultado
```

Con los resultados de resMA renombramos los parámetros de Linf, k, t0 para utilizarlos en un objeto *ModelovBA*, que es el vector de las longitudes estimadas según la edad observada.

```
LinfA = resMA$par[1]; kA = resMA$par[2]; t0A = resMA$par[3]; LinfA; kA; t0A # renombrar parámetros.
ModelovBA = (LinfA*(1-exp(-(kA*(edad-t0A))))) # datos estimados por el modelo
lines(edad, ModelovBA, lwd=2, col=1, lty=1) # línea del modelo en el gráfico
```

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \square \left(log\ (L_i) - log\ (\widehat{L}_i)\right)^2}{n}}$$

Donde: σ es la desviación de los datos, L_i: = longitud observada y $\hat{L}_i$ = longitud estimada y n el tamaño de la muestra.

Código error multiplicativo.

```
logvero  = function(pars){ # inicia creación de la función
Linf = pars[1] # asignación de L infinita
k = pars[2] # asignación de k
t0 = pars[3] # asignación del t0
sigma = pars[4] # asignación de la desviación estándar
n = length(edad) # número de observaciones
lv = 0 # es el punto de inicio de logverosimilitud
for(i in 1:n){ # inicia el loop para los n datos observados
    modelo = (Linf*(1-exp(-k*(edad[i]-t0)))) # estructura del modelo analizado
    lv = lv + log(dnorm(log(talla[i]), log(modelo), sigma)) # suma de logverosimilitud
}
-lv # se invierte la logverosimilitud, porque en R no se puede maximizar
} # Termina la estructura de la función
```

Los parámetros optimizados se conservan en el objeto resMM se renombran y se utilizan en el objeto ModelovBM donde se obtiene el vector de longitudes estimadas a partir de un modelo multiplicativo.

```
resMM = suppressWarnings(nlminb(start = c(max(talla), ifelse(max(edad) >= 10, 0.3, 0.5), -1, sd(edad)), logvero)) # suppressWarnings es para evitar caer en error.
resMM # para ver el resultado
```

Se realiza la asignación de los parámetros a su variable correspondiente.

```
LinfM = resMM$par[1]; k = resMM$par[2]; t0 = resMM$par[3]; LinfM; kM; t0M # renombrar parámetros.
ModelovBM = (LinfM*(1-exp(-(kM*(edad-t0M))))) # datos estimados por el modelo
lines(edad, ModelovBM, lwd = 2, col = 4, lty = 2) # línea del modelo en el gráfico
```

Asignación de las leyendas para diferenciar el tipo de estructura de error.

```
legend("bottomright", c("Error aditivo", "Error multiplicativo"), box.lty = 0,
       lty = 1, col = c("black", "blue"), lwd = c(2, 2))
```

Los intervalos de confianza (IC al 95%) de los parámetros en cada modelo se estiman mediante un remuestreo (bootstrap) y el método del percentil, considerando como fuente de error la talla a la edad (Haddon, 2011) para la función con estructura de *error aditivo*.

En la siguiente función el modelo realiza la estimación de los parámetros para cada remuestreo, la información del remuestreo se encuentra en una variable nombrada como *vector*.

```
ModelovBIC = (LinfA*(1-exp(-(kA*(edad-t0A))))) # datos estimados por el modelo
logvero = function(pars){
Linf = pars[1]
k =pars[2]
t0 = pars[3]
sigma = pars[4]
n = length(edad)
lv = 0
for(i in 1:n){
    modelo = (Linf*(1-exp(-k*(edad[i]-t0))))
    lv = lv + log(dnorm((vector[i]), (modelo), sigma))
}
-lv
}
```

La generación de datos en el remuestreo se realiza para 100 iteraciones en este ejercicio, pero la cantidad de remuestreo que se requiera en el análisis, se realiza cambiando el valor en *n*.

En cada uno de los remuestreos el modelo estima los parámetros y estos son almacenados en una variable nombrada como *salida*, la cual tiene una estructura de datos "*data.frame*" en R, con un orden asignado a los parámetros estimados para este ejercicio: L infinita, k, t0 y sigma.

La visualización de los parámetros estimados en cada remuestreo se puede hacer al escribir el nombre de la variable *salida*.

```
n = 1000; # cantidad de remuestreos y parámetros a estimar
salida = -data.frame(Linf = 1:n, k = 1:n, t0 = 1:n, sigma = 1:n) # nombre de los parámetros
suppressWarnings(for(i in 1:n) {# evita que converjan en un error
ERROR = c(talla - ModelovBIC) # son los errores
AB = rnorm(length(edad), mean(ERROR), sd(ERROR)) # selección aleatoria del error
para el tamaño de muestra
TBS = (ModelovBIC + AB) # tallas bootstrap
vector = TBS # reacomodo de tallas
resICA = nlminb(start=c(Linf,k,t0,sd(vector)),logvero) # Ajustes
salida[i,] = round(resICA$par[1:4],3) # Almacenamiento de parámetros con tres
decimales
  })
```

Los valores de los parámetros en la variable *salida* están orden aleatorio y para estimar los intervalos de confianza se ordenan de menor a mayor y los intervalos de confianza al 95%, se encuentran en el valor 25 y 975 respectivamente.

```
##### Valores de los intervalos de confianza de los parámetros #####
# L infinita
icL=salida$Linf[order(salida$Linf)]# ordena los datos de menor a mayor.
LqL <- icL[(n+1) * 25/1000]# inferior
LqS <- icL[(n+1) * 975/1000]# superior
# Tasa de crecimiento k
ick=salida$k[order(salida$k)]# ordena los datos de menor a mayor.
kqL <- ick[(n+1) * 25/1000]# inferior
kqS <- ick[(n+1) * 975/1000]# superior
# Edad teorica a la talla cero
ict0=salida$t0[order(salida$t0)]# ordena los datos de menor a mayor.
t0qL <- ict0[(n+1) * 25/1000]# inferior
t0qS <- ict0[(n+1) * 975/1000]# superior
##### Tallas a la edad para los intervalos
ModelovBICL = (LqL*(1-exp(-(kqL*(edad-t0qL))))) # datos estimados por el modelo
ModelovBICS = (LqS*(1-exp(-(kqS*(edad-t0qS))))) # datos estimados por el modelo
##### Grafico de tallas para los intervalos
lines(edad,ModelovBICL,lwd=2,col=4,lty=2)
lines(edad,ModelovBICS,lwd=2,col=4,lty=2)
###### Leyenda de datos
legend("bottomright", c("Modelo", "IC al 95%"), box.lty = 0,
       lty = c(1,2), col = c("black", "blue"), lwd = c(2, 2))
LqL;LqS;kqL;kqS;t0qL;t0qS
```

RESULTADOS

Se muestra el modelo de von Bertalanffy ajustado a los datos de Redfish *Centroberyx affinis* para estructura de error aditivo y multiplicativo (figura 2). Los parámetros estimados se obtienen al llamar la variable resMA para la estructura de error aditivo (considere el cambio de nombre en la variable a resMM para obtener el resultado correcto con error multiplicativo), los parámetros de interés de resMA se encuentran en el primer reglón en la sección $par donde: resMA$par[1] es Linf, resMA$par[2] corresponde a k, resMA$par[3] es t0 y resMA$par[4] a sigma. La logverosimilitud se encuentra en resMA$objetive.

```
$par
[1] 26.8354017  0.1301586 -3.5867046  1.9500909
$objective
[1] 747.0795
$convergence
[1] 0
$iterations
[1] 29
$evaluations
function gradient
      50      138
$message
[1] "relative convergence (4)"
```

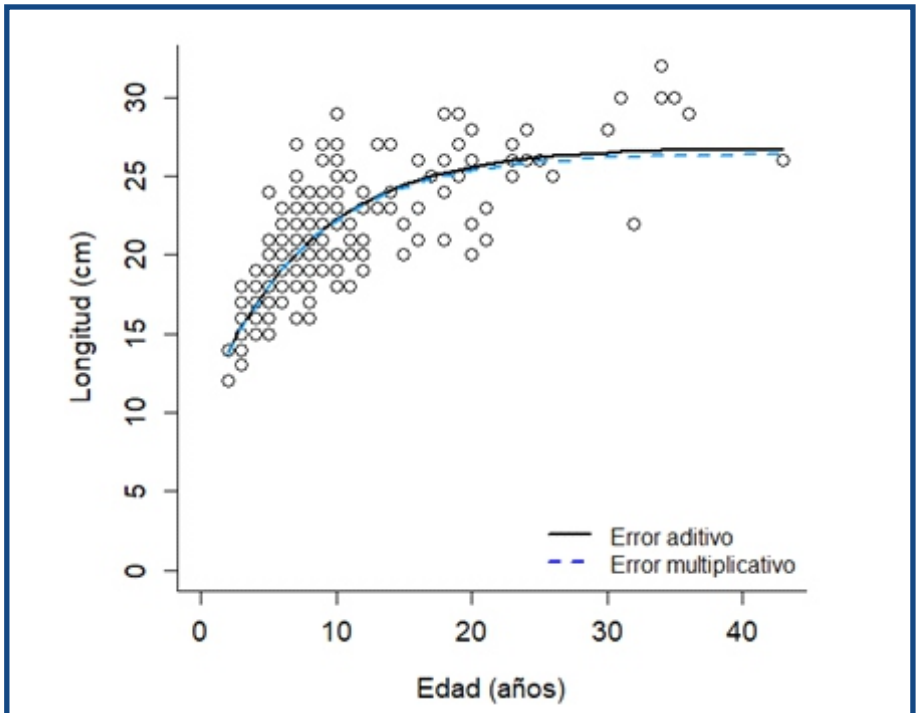

Figura 2. Modelo de von Bertalanffy ajustado a los datos de Redfish, *Centroberyx affinis* (Haddon, 2021), línea negro error aditivo y línea azul error multiplicativo.

Si se quiere replicar el resto de los ejercicios es necesario copiar y pegar los datos según sea el caso y ejecute el código.

La comparación de los parámetros estimados del modelo de von Bertalanffy con el código propuesto y los estimados con los repositorios virtuales se encuentran en la tabla I, en esta se observa una coincidencia en los valores estimados entre las estructuras de error independientemente del método de ajuste.

Tabla I. Comparativo de los parámetros, estructuras de error (EA = error aditivo y EM = multiplicativo) y método de ajuste (LL = logverosimilitud, MC = mínimos cuadrados).

Datos	Parámetro	L_∞	k	t0	Error	Ajuste
Centroberyx affinis	Este estudio	26.835	0.130	-3.586	EA	LL
	Haddon, 2021	26.835	0.130	-3.586	EA	MC
	Este estudio	26.441	0.137	-3.294	EM	LL
	Haddon, 2021	26.441	0.137	-3.294	EM	MC
Pogonias cromis	Este estudio	1196.719	0.141	-1.594	EA	LL
	Ogle, 2016	1196.718	0.141	-1.594	EA	MC
Lagodon rhomboides	Este estudio	211.785	0.378	-0.915	EA	LL
	Nelson, 2023	211.787	0.378	-0.915	EA	MC
Lagodon rhomboides	Este estudio	190.396	0.537	-0.537	EA	LL
	Nelson, 2023	190.398	0.537	-0.537	EM	MC
Merluccius productus (hembras)	Este estudio	61.225	0.296	-0.057	EA	LL
	Kimura 1981	61.23	61.23	–0.057	EA	MC
Merluccius productus (machos)	Este estudio	55.978	0.385	0.171	EA	LL
	Kimura 1981	55.978	0.385	0.171	EA	MC
Istiophorus platypterus	Este estudio	180.588	0.363	-0.241	EM	LL
	Cerdenares-Ladrón De Guevara et al., 2011	180.588	0.363	-0.241	EM	LL

El modelo ajustado a cada conjunto de datos analizados se encuentra en la figura 3, se observa que independientemente de la dispersión de los datos analizados, el código optimiza los parámetros del modelo. En la tabla II, se encuentran los parámetros del modelo y sus intervalos de confianza.

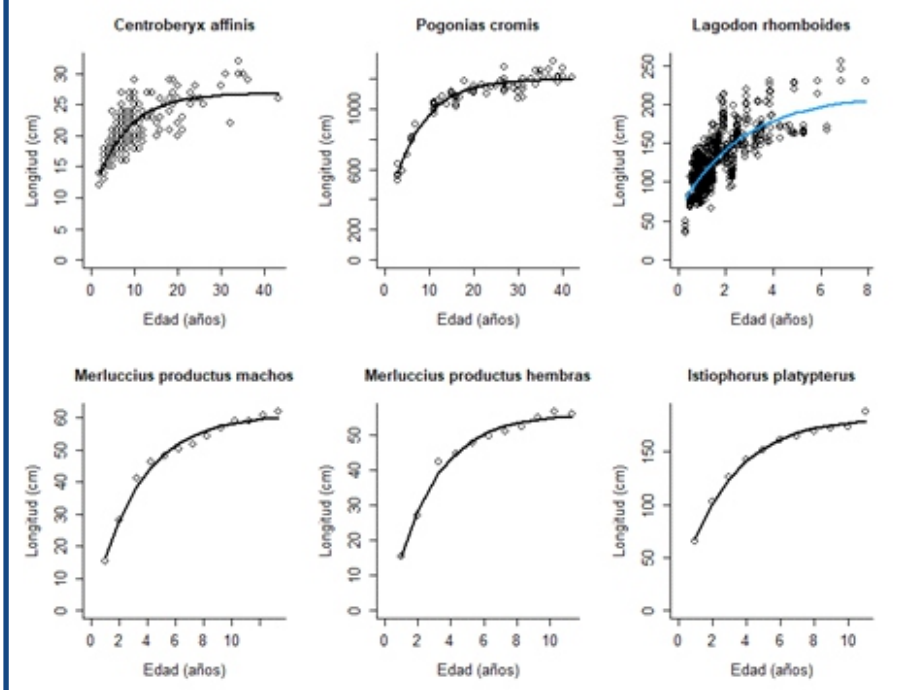

Figura 3. Modelo de von Bertalanffy ajustado a los datos de los repositorios virtuales error aditivo.

Datos	Parámetro	L_∞	k	t0	Ajuste
Centroberyx affinis	Estimación	26.835	0.130	-3.586	Error aditivo
	IC 95% LI	25.87	0.107	-4.911	
	IC 95% LS	27.914	0.157	-2.483	
Pogonias cromis	Estimación	1196.719	0.141	-1.594	Error aditivo
	IC 95% LI	1179.107	0.123	-2.606	
	IC 95% LS	1214.346	0.162	-0.831	
Lagodon rhomboides	Estimación	211.785	0.378	-0.915	Error aditivo
	IC 95% LI	196.172	0.272	-1.319	
	IC 95% LS	236.854	0.485	-0.63	
Merluccius productus (machos)	Estimación	61.225	0.296	-0.057	Error aditivo
	IC 95% LI	59.148	0.248	-0.403	
	IC 95% LS	63.625	0.35	0.224	
Merluccius productus (hembras)	Estimación	55.978	0.385	0.171	Error aditivo
	IC 95% LI	54.131	0.326	-0.08	
	IC 95% LS	58.079	0.457	0.401	
Istiophorus platypterus	Estimación	180.588	0.363	-0.241	Error multiplicativo
	IC 95% LI	175.512	0.304	-0.594	
	IC 95% LS	186.804	0.43	0.012	

Tabla II. Parámetros del modelo de von Bertalanffy, intervalos de confianza (95%) y tipo de ajuste.

En la figura 4 se observa el modelo de von Bertalanffy ajustado con error aditivo y sus respectivos intervalos de confianza.

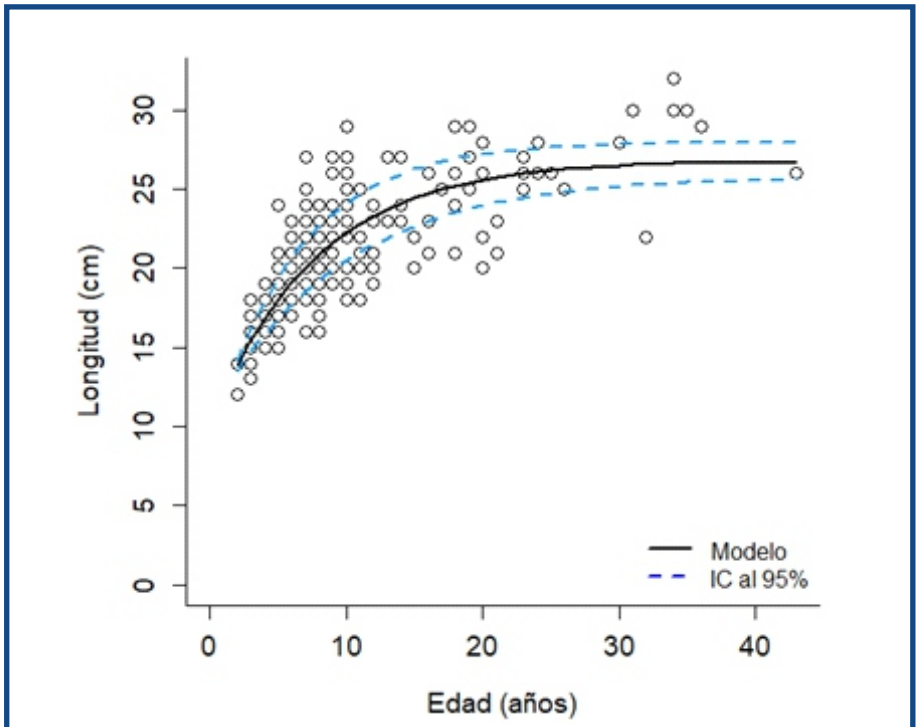

Figura 4. Modelo de von Bertalanffy ajustado a los datos de Redfish, *Centroberyx affinis* (Haddon, 2021), línea negra el modelo y línea azul los intervalos de confianza al 95%.

DISCUSIÓN

El ajuste de un modelo de crecimiento individual en Microsoft Excel, se realiza relativamente fácil cuando se replica un ejercicio tipo (Haddon, 2011). La bondad de Microsoft Excel estriba en que el modelo se escribe con los parámetros fijos en la primera celda del ejercicio y posteriormente con un doble click se estiman el resto de las celdas, lo cual no sucede en R, pero esto se resuelve de una mejor forma en R con la aplicación del término estadístico "i-esimo" en la función, la cual considera a todos los datos analizados independientemente de la cantidad, sin necesidad de editar como sucede en Microsoft Excel.

En ese contexto en R solo necesita agregar los datos en variables nombradas edad y talla, ejecutar el código y obtiene de forma estándar los resultados del análisis.

El interés de realizar el ejercicio de ajustar un modelo de crecimiento individual en R, sin el uso de una función predeterminada como nls, se debe a que esta realiza el ajuste por mínimos cuadrados y la función *nlminb()* ajusta al modelo por verosimilitud, mediante el cual se pueden estimar los intervalos de confianza con pruebas estadísticas para los parámetros del modelo (Polacheck, Hilborn, Punt, 1993), pero además, la función madre es un código donde se observa y describe cada proceso, como se realiza en Microsoft Excel.

También existe otras funciones de optimización disponibles en R como *nlm()* u *optim()* (Haddon, 2021), la cuales difieren en la estructura o ingreso de los parámetros y función objetivo. Pero el uso de *nlminb()* estimó los valores de los parámetros que permitieron validar la función en todos los conjuntos de datos, lo cual no sucedió con *nlm()*, mientras que en el caso de *optim()* los parámetros difieren según el método de ajuste (por default usa el método de Nelder-Mead).

La herramienta solver de Microsoft Excel tiene la opción de maximizar o minimizar la función objetivo y se elige según se requiera. Cuando se optimizan los parámetros de un modelo la verosimilitud negativa y positiva se minimizan y maximizan respectivamente. La diferencia en la función objetivo es el signo pero el valor absoluto es el mismo.

En R solo tiene la opción de minimizar, pero en la función logvero se realiza la suma de la verosimilitud positiva, por esa razón se invierte el signo al final de la función, de lo contrario sería imposible optimizar los parámetros y la función indicaría un error.

Se recuerda que el signo de logvero se cambió a negativo para minimizar (R no maximiza), si se quiere comparar el ajuste entre estructura de error en el modelo, usando un método de selección de modelo, debe cambiarse el valor de logvero (resMA$objective o resMM$objective) al multiplicar por menos uno (-1), ya que los métodos de selección utilizan la máxima verosimilitud (Burnham y Anderson, 2002).

El uso de los datos edad-talla en repositorios virtuales (Kimura, 1980; Ogle, 2016; Haddon, 2021; Nelson, 2023) ajustados a un modelo con estructura de error aditivo, ayudó a evaluar la estructura de la función *logvero*, en la cual coinciden los valores de los parámetros estimados y la verosimilitud, pero creamos un segundo código de la función para una estructura de error multiplicativo y se validó con los datos en repositorios virtuales (Haddon, 2021; Nelson, 2023) y datos del pez vela *Istiophorus platypterus* (Cerdenares-Ladrón De Guevara et al., 2011).

La diferencia en ambas funciones es el uso de logaritmo natural en los datos (ver función), al ajustar un modelo en Microsoft Excel con error multiplicativo, los errores (residuales) se obtienen como logaritmo de una división elevada al cuadrado, entonces para resolver esa parte se aplicó la *ley de los logaritmos* "log(talla[i]), log(modelo)", ya que las divisiones equivalen a restas y las multiplicaciones a sumas.

Al optimizar los parámetros en un modelo es importante considerar la certidumbre de su estimación, en ese contexto se elaboró el método de estimación de los intervalos de confianza mediante de una técnica de remuestreo, realizada en visual base (macro) Microsoft Excel (Haddon, 2011) y se escribió en lenguaje de programación R.

En el proceso se estimaron los errores de las tallas con respecto al modelo para un tamaño de muestra de *n* (Haddon, 2021), asumiendo que se distribuyen normalmente cómo en la estructura del error en el ajuste del modelo. Este remuestreo se realiza *n* veces y los parámetros optimizados se encuentran en una variable nombrada *salida,* finalmente se obtienen los intervalos de confianza para cada parámetro con la función de los 1,000 datos ordenados de menor a mayor, donde el valor 25 y 975 son los intervalos de confianza inferior y superior respectivamente.

El ejercicio de optimización de los parámetros con la función propuesta muestra buen desempeño, al coincidir en las estimaciones de los parámetros del modelo, al utilizar los datos de los repositorios virtuales (Kimura, 1980; Cerdenares-Ladrón De Guevara et al., 2011; Ogle, 2016; Haddon, 2021; Nelson, 2023), por lo que se recomienda su uso para la estimación de los parámetros de un modelo de crecimiento individual, adaptando la *función madre* según sea el modelo y el error que queramos asumir.

BIBLIOGRAFÍA

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: a practical information-theoretic approach* (2nd ed.). Springer.

Cerdenares-Ladrón De Guevara, G., Morales-Bojórquez, E., & Rodríguez-Sánchez, R. (2011). Age and growth of the sailfish Istiophorus platypterus (Istiophoridae) in the Gulf of Tehuantepec, Mexico. *Marine Biology Research*, *7*(5), 488-499. https://doi.org/10.1080/17451000.2010.528201

Genschel, U. y Meeker, W.Q. (2010). A Comparison of Maximum Likelihood and Median-Rank Regression for Weibull Estimation. *Quality Engineering*, 22(4): 236–255.

Haddon, M. (2011). *Modelling and Quantitative Methods in Fisheries* (2nd ed.). Chapman and Hall; CRC Press. https://doi.org/https://doi.org/10.1201/9781439894170

Haddon, M. (2021). *Using R for Modelling and Quantitative Methods in Fisheries*. Chapman and Hall; CRC Press. https://doi.org/10.1201/9781003032601

Kimura, D. K. (1980). Likelihood methods for the von Bertalanffy growth curve. *Fishery Bulletin*, *77*, 765-776.

Nelson, G.A. (2023). Fishery Science Methods and Models. Repository CRAN. https://cran.r-project.org/web/packages/fishmethods/fishmethods.pdf

Ogle, D. (2016). *Introductory Fisheries Analyses with R*. Chapman and Hall; CRC Press. https://doi.org/https://doi.org/10.1201/9781315371986

Polacheck T, R Hilborn y AE Punt. 1993. Fitting surplus production models: Comparing methods and measuring uncertainty. Canadian Journal of Fisheries and Aquatic Sciences 50(12): 2597-2607.

Posit Team. (2023). *R: A language and environment for statistical computing*. In R Foundation for Statistical Computing.

von Bertalanffy, L. (1938). A quantitative theory of organic growth. *Human Biology*, *10*(2), 181-213.