

# Función “PBR\_F”

22 Junio, 2016

**Lenguaje:** MATLAB

**Versión:** 8.1.0.604 (R2013a)

**Autor:** Manuela Mendiolar < [mmendiolar@inidep.edu.ar](mailto:mmendiolar@inidep.edu.ar) >

**Marco:** Proyecto FISH-GURU+ - Gabinete de Biomatemática - INIDEP

**Descripción:** Función que estima  $\lambda_{max}$  y  $\lambda_{0.1}$  del Análisis de Rendimiento por Recluta de Thompson y Bell, teniendo en cuenta la incertidumbre en los pesos medios por edad, tasas instantáneas de mortalidad natural por edad y tasas instantáneas de mortalidad por pesca por edad.

**Plataformas:** MATLAB está disponible para Unix, Windows, Mac OS X y GNU/Linux .

## Contenidos

Sintaxis .....	2
Argumentos.....	2
Descripción.....	2
Ejemplo.....	3
Representación gráfica.....	4
Código PBR_F .....	5
Referencias.....	5

## Sintaxis

[x<sub>max</sub>, x<sub>01</sub>] = PBR\_F(F, M, W, x<sub>0</sub>)

## Argumentos

### Entradas

- F : vector de tasas instantáneas de mortalidad por pesca por edad (*vector fila*).
- M : vector de tasas instantáneas de mortalidad natural por edad (*vector fila*).
- W : vector de pesos medios por edad (*vector fila*).
- x<sub>0</sub> : valor inicial para la resolución de la ecuación no lineal para estimar  $\lambda_{0.1}$ .

### Salidas

- x<sub>max</sub> : valor donde la función rendimiento por recluta alcanza el máximo ( $\lambda_{max}$ ).
- x<sub>01</sub> : valor donde la derivada de la función rendimiento por recluta equivale a (1/10) de la derivada de la función rendimiento por recluta en el origen ( $\lambda_{0.1}$ ).

## Descripción

La función **PBR\_F** calcula los Puntos Biológicos de Referencia (PBR)  $\lambda_{max}$  y  $\lambda_{0.1}$  del Análisis de Rendimiento por Recluta (Y/R, por sus siglas en inglés) de Thompson y Bell [\[1\]](#), dados los pesos medios por edad, las tasas instantáneas de mortalidad natural por edad y las tasas instantáneas de mortalidad por pesca por edad.

La función Y/R *en peso* suele expresarse en función de un sólo parámetro:

$$Y/R(\lambda) = \frac{\lambda F_1}{\lambda F_1 + M_1} (1 - e^{-\lambda F_1 - M_1}) W_1 + \sum_{i=2}^{k-1} \left[ \frac{\lambda F_i}{\lambda F_i + M_i} (1 - e^{-\lambda F_i - M_i}) W_i \prod_{j=1}^{i-1} e^{-\lambda F_j - M_j} \right] + \frac{\lambda F_{k+}}{\lambda F_{k+} + M_{k+}} W_{k+} \prod_{j=1}^{k-1} e^{-\lambda F_j - M_j}$$

donde:

$F_i$  : es la tasa instantánea de mortalidad por pesca para la edad  $i$

$M_i$  : es la tasa instantánea de mortalidad natural para la edad  $i$

$W_i$  : es el peso medio para la edad  $i$

$k +$ : grupo plus (agrupa las edades  $k, k + 1, k + 2, \dots, k_{max}$ )

El parámetro  $\lambda \geq 0$  es un factor de escala que mide el esfuerzo empleado. Esta función puede utilizarse luego para calcular los vectores  $F_{max}$  y  $F_{0.1}$ , a partir de los valores  $\lambda_{max}$  y  $\lambda_{0.1}$  que se definen como:

- ◆  $\lambda_{max}$  valor donde la función Y/R alcanza el máximo.
- ◆  $\lambda_{0.1}$  valor donde la derivada de la función Y/R es igual (1/10) de la derivada de la función Y/R evaluada en el origen. Este valor se calcula resolviendo la ecuación no lineal:

$$\left. \frac{d(Y/R)}{d\lambda} \right|_{\lambda=\lambda_{0.1}} = \frac{1}{10} \left. \frac{d(Y/R)}{d\lambda} \right|_{\lambda=0} \quad (1)$$

Una vez obtenidos  $\lambda_{max}$  y  $\lambda_{0.1}$ , los vectores  $F_{max}$  y  $F_{0.1}$  se calculan como:

- $F_{max} = \lambda_{max} F$
- $F_{0.1} = \lambda_{0.1} F$

**Nota:**

Para estimar  $\lambda_{max}$ , **PBR\_F** utiliza la función **fminbnd** cuyo algoritmo se basa en la búsqueda de sección áurea y la interpolación parabólica. Para más detalle sobre el algoritmo véase [2] y [3]. Para resolver la ecuación no lineal (1), **PBR\_F** utiliza la función **fzero**. Esta última emplea una combinación de los métodos numéricos de bisección, secante e interpolación cuadrática inversa. La versión Fortran, en la cual está basada **fzero** puede verse en [4].

## Ejemplo

Para el siguiente ejemplo sencillo con **PBR\_F** se emplearon datos que corresponden a la especie merluza (*Merluccius hubbsi*), clase anual 1998 [5].

```
F = [0.055, 0.893, 1.271, 1.154, 0.900, 0.920, 0.920];
M = [0.300, 0.300, 0.300, 0.300, 0.300, 0.300, 0.300];
W = [0.164, 0.337, 0.577, 0.906, 1.232, 1.581, 1.939]; % (Kg.)

x0 = 0.3; % valor inicial (una manera de elegir este valor podría ser graficando
previamente la función RPR)

[xmax, x01] = PBR_F(F, M, W, x0)
```

```
xmax =

    0.4899

x01 =

    0.2489
```

Se pueden calcular los vectores  $F_{max}$  y  $F_{0.1}$ :

```
Fmax = xmax*F
F01 = x01*F
```

```
Fmax =

    0.0269    0.4375    0.6227    0.5654    0.4409    0.4507    0.4507
```

F01 =

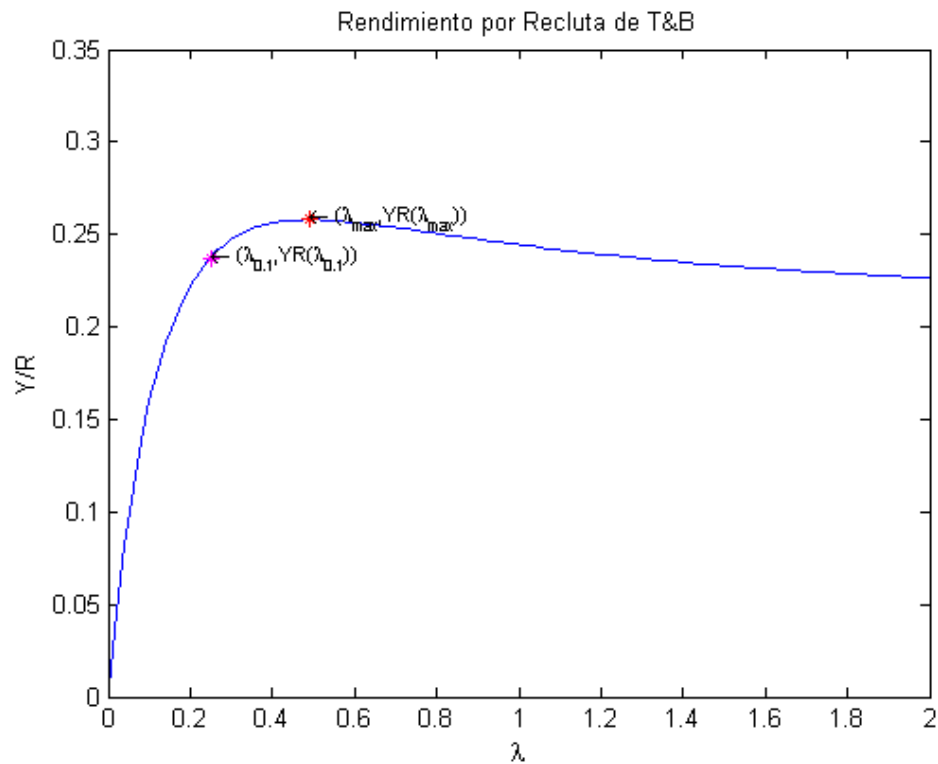
0.0137    0.2222    0.3163    0.2872    0.2240    0.2289    0.2289

## Representación gráfica

```
xdata = linspace(0, 2, 100);
for i=1: 100,
    ydata(i) = YR_TB(xdata(i), F, M, W);
end

plot(xdata, ydata, 'b', 0.4899, 0.2579, 'r*', 0.2489, 0.2374, 'm*', 'LineWidth', 1);
xlabel('\lambda');
ylabel('YR(\lambda)');
title('Rendimiento por Recluta de T&B');

% Marcamos los puntos (0.4899, 0.2579) y (0.2489, 0.2374)
text(0.4899, 0.2579, '\leftarrow'
(\lambda_{max}, YR(\lambda_{max})), 'Horizontal Alignment', 'left', 'FontSize', 8);
text(0.2489, 0.2374, '\leftarrow'
(\lambda_{0.1}, YR(\lambda_{0.1}))), 'Horizontal Alignment', 'left', 'FontSize', 8);
```



## Código PBR\_F

```
function [xmax, x01] = PBR_F(F, M, W, x0)

f1 = @(x) NYR_TB(x);
f2 = @(x) (dYR_TB(x) - 0.1*dYR_TB(0));

xmax = fminbnd(@(x) f1(x), 0, 10);
x01 = fzero(f2, x0);

function y = NYR_TB(x) % (-Y/R)
    p = exp(-(x.*F) - M);
    G = (x.*F) ./ ((x.*F) + M);
    k = length(F);
    aux = 1;
    sum = 0;
    for i=1:k-1,
        sum = sum + G(i)*(1-p(i))*aux*W(i);
        aux = aux*p(i);
    end
    sum = sum + G(k)*aux*W(k);
    y = -sum;
end

function y = dYR_TB(x) % (Y/R)'
    p = exp(-(x.*F) - M);
    G = (x.*F) ./ ((x.*F) + M);
    dG = (F.*M) ./ (((x.*F)+M) .* ((x.*F)+M));
    daux = 1;
    dsum = 0;
    sumF = 0;
    k = length(F);
    for i=1:k-1,
        dsum = dsum + (G(i)*F(i)*p(i) + (dG(i) - G(i)*sumF)*(1-p(i)))*W(i)*daux;
        sumF = sumF + F(i);
        daux = daux*p(i);
    end
    dsum = dsum + (dG(k) - G(k)*sumF)*W(k)*daux;
    y = dsum;
end
```

## Ver también

**YR\_TB** (función que calcula el rendimiento por recluta de Thompson y Bell en peso  $Y/R$ ).

## Referencias

- [1] Thompson, W. F. y Bell, F. H. Biological statistics of the Pacific halibut fishery: (2) effect of changes in intensity upon total yield and yield per unit of gear International Fisheries Commission, 1934.

- [2] Forsythe, G. E., M. A. Malcolm, and C. B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, 1976.
- [3] Brent, Richard. P., Algorithms for Minimization without Derivatives, Prentice-Hall, Englewood Cliffs, New Jersey, 1973
- [4] Forsythe, G. E., M. A. Malcolm, y C. B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, 1976.
- [5] Irusta, C. G., Evaluación del estado del efectivo norte de 41°S de la merluza (*Merluccius hubbsi*) y estimación de la captura biológicamente aceptable para el año 2016, Inf. Tec. Int. INIDEP N°29, 2015.

*Published with MATLAB® R2013a*