

# MAT-466: Modelo de regresión no lineal

**Felipe Osorio**

<http://fosorios.mat.utfsm.cl>

Departamento de Matemática, UTFSM



# Modelos de regresión no lineal

Considere el conjunto de **ecuaciones de regresión**:

$$Y_i = f(\mathbf{x}_i; \boldsymbol{\theta}) + \epsilon_i, \quad i = 1, \dots, n, \quad (1)$$

donde  $Y_i$  denota las respuestas observadas,  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top$  es vector de variables regresoras  $p \times 1$ ,  $f(\mathbf{x}; \boldsymbol{\theta})$  es función conocida, mientras que  $\boldsymbol{\theta}$  es vector  $p$ -dimensional de parámetros desconocidos, y  $\epsilon_i$  representa disturbios aleatorios.

El modelo en (1) puede ser escrito de forma conveniente como:

$$\mathbf{Y} = \mathbf{f}(\boldsymbol{\theta}) + \boldsymbol{\epsilon},$$

donde

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{f}(\boldsymbol{\theta}) = \begin{pmatrix} f(\mathbf{x}_1; \boldsymbol{\theta}) \\ \vdots \\ f(\mathbf{x}_n; \boldsymbol{\theta}) \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$



Además, consideraremos los siguientes **supuestos de momentos**:

$$E(\epsilon) = \mathbf{0}, \quad \text{Cov}(\epsilon) = \sigma^2 \mathbf{I}_n,$$

y defina:

$$S(\boldsymbol{\theta}) = \sum_{i=1}^n \{Y_i - f(\mathbf{x}_i; \boldsymbol{\theta})\}^2 = \|\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta})\|^2.$$

El objetivo de la siguiente sección será introducir un procedimiento para obtener  $\hat{\boldsymbol{\theta}}$  como:

$$\hat{\boldsymbol{\theta}} := \arg \min_{\boldsymbol{\theta}} S(\boldsymbol{\theta})$$



### Ejemplo:

En demografía se ha usado modelos de tipo 'gravedad' para caracterizar la migración,

$$I = \frac{kP_1P_2}{D^b},$$

donde  $k$  y  $b$  son constantes,  $I$  representa el nivel de migración (interacción) entre dos áreas a distancia  $D$  con poblaciones  $P_1$  y  $P_2$ , respectivamente.

Note que

$$\log \left( \frac{I}{P_1P_2} \right) = \log k - b \log D = \beta_1 + \beta_2 x.$$



## Ejemplo:

En una reacción química irreversible, una sustancia  $A$  cambia a la sustancia  $B$ , la que cambia a la sustancia  $C$ . Sea  $A(t)$  la cantidad de sustancia  $A$  en el tiempo  $t$ . Las ecuaciones que rigen tales cambios son dadas por:

$$\begin{aligned}\frac{dA(t)}{dt} &= -\theta_1 A(t), \\ \frac{dB(t)}{dt} &= \theta_1 A(t) - \theta_2 B(t), \\ \frac{dC(t)}{dt} &= \theta_2 B(t),\end{aligned}$$

donde  $\theta_1$  y  $\theta_2$  son constantes. Asumiendo  $A(0) = 1$ ,  $B(t)$  tiene solución:

$$B(t) = \frac{\theta_1}{\theta_1 - \theta_2} \{ \exp(-\theta_2 t) - \exp(-\theta_1 t) \}.$$



## Mínimos cuadrados no lineales: Método Gauss-Newton

Estimadores usados en **regresión no lineal** pueden ser caracterizados como **formas lineales** y **cuadráticas** bastante similares a las que surgen en regresión lineal.

En efecto, sea

$$\mathbf{F}(\boldsymbol{\theta}) = \frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\top} = \left( \frac{\partial f(\mathbf{x}_i; \boldsymbol{\theta})}{\partial \theta_j} \right).$$

Note por otro lado que:

$$\begin{aligned} \frac{\partial S(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} (\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta}))^\top (\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta})) = 2 \left( \frac{\partial}{\partial \boldsymbol{\theta}} (\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta})) \right)^\top (\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta})) \\ &= -2\mathbf{F}^\top(\boldsymbol{\theta})(\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta})). \end{aligned}$$

De modo que,  $\hat{\boldsymbol{\theta}}$  es solución de la **ecuación de estimación**:

$$\boldsymbol{\Psi}(\boldsymbol{\theta}) = \mathbf{F}^\top(\boldsymbol{\theta})(\mathbf{Y} - \mathbf{f}(\boldsymbol{\theta})).$$



## Mínimos cuadrados no lineales: Método Gauss-Newton

Considere una estimación inicial  $\theta^{(0)}$  y considere la **aproximación de primer orden**:

$$\mathbf{f}(\theta) \approx \mathbf{f}(\theta^{(0)}) + \mathbf{F}(\theta^{(0)})(\theta - \theta^{(0)}),$$

de este modo,

$$\begin{aligned} S(\theta) &= \|\mathbf{Y} - \mathbf{f}(\theta)\|^2 \\ &\approx \tilde{S}(\theta) \\ &= \|\mathbf{Y} - \mathbf{f}(\theta^{(0)}) - \mathbf{F}(\theta^{(0)})(\theta - \theta^{(0)})\|^2 \\ &= \|\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\theta - \theta^{(0)})\|^2, \end{aligned}$$

donde  $\mathbf{e}^{(0)} = \mathbf{Y} - \mathbf{f}(\theta^{(0)})$  y  $\mathbf{F}^{(0)} = \mathbf{F}(\theta^{(0)})$ .

*Observación:*

Note que,  $\tilde{S}(\theta)$  es una **función cuadrática** en  $\theta$ .



## Mínimos cuadrados no lineales: Método Gauss-Newton

Diferenciando  $\tilde{S}(\boldsymbol{\theta})$  con relación a  $\boldsymbol{\theta}$  obtenemos,

$$\begin{aligned}\frac{\partial \tilde{S}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= 2 \left( \frac{\partial}{\partial \boldsymbol{\theta}} (\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})) \right)^{\top} (\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})) \\ &= -2 \mathbf{F}^{(0)\top} (\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})),\end{aligned}$$

igualando a cero y resolviendo, obtenemos que para  $\mathbf{F}^{(0)}$  matriz de rango completo,  $\tilde{S}(\boldsymbol{\theta})$  es mínimo para

$$\mathbf{F}^{(0)\top} \mathbf{F}^{(0)} (\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}) = \mathbf{F}^{(0)\top} \mathbf{e}^{(0)},$$

es decir

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(0)} + (\mathbf{F}^{(0)\top} \mathbf{F}^{(0)})^{-1} \mathbf{F}^{(0)\top} \mathbf{e}^{(0)}.$$





## Mínimos cuadrados no lineales: Método Gauss-Newton

Diferenciando  $\tilde{S}(\boldsymbol{\theta})$  con relación a  $\boldsymbol{\theta}$  obtenemos,

$$\begin{aligned}\frac{\partial \tilde{S}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= 2 \left( \frac{\partial}{\partial \boldsymbol{\theta}} (\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})) \right)^{\top} (\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})) \\ &= -2 \mathbf{F}^{(0)\top} (\mathbf{e}^{(0)} - \mathbf{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})),\end{aligned}$$

igualando a cero y resolviendo, obtenemos que para  $\mathbf{F}^{(0)}$  matriz de rango completo,  $\tilde{S}(\boldsymbol{\theta})$  es mínimo para

$$\mathbf{F}^{(0)\top} \mathbf{F}^{(0)} (\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}) = \mathbf{F}^{(0)\top} \mathbf{e}^{(0)},$$

es decir

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(0)} + (\mathbf{F}^{(0)\top} \mathbf{F}^{(0)})^{-1} \mathbf{F}^{(0)\top} \mathbf{e}^{(0)}.$$



Finalmente obtenemos la iteración

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{\delta}^{(k)}, \quad k = 0, 1, \dots$$

donde el incremento  $\boldsymbol{\delta}^{(k)}$  es solución del sistema de ecuaciones lineales (sobredeterminadas)

$$\mathbf{F}^{(k)} \boldsymbol{\delta}^{(k)} = \mathbf{e}^{(k)}.$$

algoritmo que es conocido como **método Gauss-Newton**.

### *Observación:*

Note que  $\boldsymbol{\delta}^{(k)}$  corresponde a un problema LS con solución:

$$\boldsymbol{\delta}^{(k)} = (\mathbf{F}^{(k)\top} \mathbf{F}^{(k)})^{-1} \mathbf{F}^{(k)\top} \mathbf{e}^{(k)}.$$



Para asegurar que el incremento de Gauss,  $\delta^{(k)}$  produce **direcciones de descenso**, se puede realizar una reducción de paso (step-halving) tal que

$$S(\theta^{(k)} + \lambda \delta^{(k)}) < S(\theta^{(k)}),$$

donde  $\lambda \in (0, 1]$  es el largo de paso.



# Datos de Puromycin (Treolar, 1974)<sup>1</sup>

**Modelo Michaelis-Menten:** usado para el estudio de cinética de enzimas.

Permite estudiar la relación entre *velocidad inicial* de una reacción enzimática a la concentración de un sustrato  $x$  a través de la ecuación:

$$f(x, \theta) = \frac{V_m x}{K + x}, \quad \theta = (V_m, K)^\top.$$

Diferenciando  $f$  con relación a  $V_m$  y  $K$ , obtenemos

$$\frac{\partial f}{\partial V_m} = \frac{x}{K + x}, \quad \frac{\partial f}{\partial K} = -\frac{V_m x}{(K + x)^2}$$

Note que el modelo Michaelis-Menten puede ser transformado en un modelo lineal, como

$$\frac{1}{f} = \frac{1}{V_m} + \frac{K}{V_m} \frac{1}{x} = \beta_1 + \beta_2 z.$$

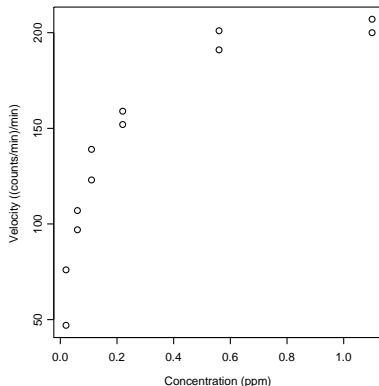
---

<sup>1</sup>Master's dissertation. University of Toronto.



## Datos de Puromycin (Treolar, 1974)

Velocidad de reacción versus concentración del sustrato para un experimento de Puromycin



# Ventajas del modelo de regresión no lineal

- ▶ Más **flexibilidad** en la construcción/elección del modelo.
- ▶ la función de modelo se basa en la **teoría** respecto del mecanismo que produce la respuesta (**no es un modelo empírico**).
- ▶ **predicciones** fuera del rango de observación puede ser realizadas con **mayor confianza**.
- ▶ Parámetros tiene un **significado** físico y por tanto son de interés primario.



## Desventajas del modelo de regresión no lineal

- ▶ Estimación de parámetros no tiene forma explícita. Se debe utilizar **algoritmos iterativos**.
- ▶ Necesidad de proveer **estimaciones iniciales** para el procedimiento de estimación.
- ▶ Errores estándar, intervalos de confianza, sólo son **aproximados**. Mayor precisión en la variabilidad de estimaciones requiere de **mayor esfuerzo computacional**.
- ▶ Posibilidad de óptimos **múltiples** y/o falsa **convergencia**.

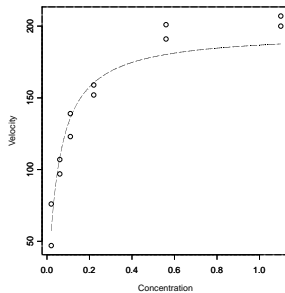
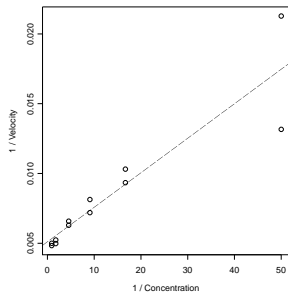


- ▶ Transformación de los datos también involucra transformar los **disturbios aleatorios**.
- ▶ Parámetros transformados **carecen de interpretación**.
- ▶ Por supuesto, una gran cantidad de modelos de interés **no son linealizables**.





# Datos de Puromycin



Realizando el ajuste en el **modelo linealizado** (usando mínimos cuadrados), obtenemos

$$\hat{\beta} = (0.005107, 0.000247)^{\top},$$

luego

$$\hat{\theta} = (195.80, 0.04841)^{\top},$$

Mientras que, usando **minimos cuadrados no lineales** obtuvimos<sup>2</sup>

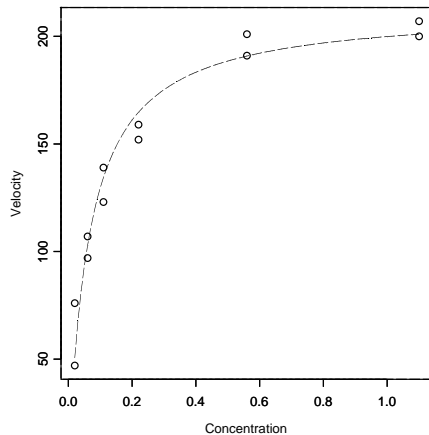
$$\hat{\theta} = (212.68, 0.06412)^{\top}.$$

---

<sup>2</sup>Se consideró  $\theta^{(0)} = (205, 0.08)^{\top}$



## Datos de Puromycin



# Datos de Puromycin (Treolar, 1974)

## Conjunto de datos

```
1 > puromycin <- list(  
2 + conc = c(.02, .02, .06, .06, .11, .11, .22, .22, .56, .56, 1.10,  
3 + vel = c(76, 47, 97, 107, 123, 139, 159, 152, 191, 201, 207, 200))  
4  
5 > puromycin <- as.data.frame(puromycin)
```

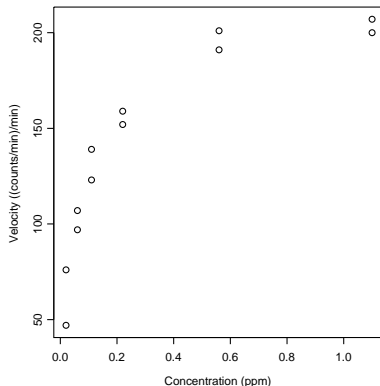
Exploramos el conjunto de datos por medio del gráfico:

```
1 > plot(vel ~ conc, data = puromycin,  
2 + xlab = "Concentration (ppm)",  
3 + ylab = "Velocity ((counts/min)/min)")
```



## Datos de Puromycin (Treolar, 1974)

Velocidad de reacción versus concentración del sustrato para un experimento de Puromycin



# Ajuste de los datos de Puromycin

Ajuste usando la función `nls()`:

```
1  > fm1 <- nls(vel ~ Vm * conc / (K + conc), data = puromycin,  
2  +          start = list(K = 0.08, Vm = 205), trace = TRUE)  
3  3155.004 :      0.08 205.00  
4  1205.662 :      0.06289222 213.02889453  
5  1195.477 :      0.06398775 212.60337573  
6  1195.449 :      0.06410830 212.67543440  
7  1195.449 :      0.06412003 212.68293989  
8  1195.449 :      0.06412116 212.68366582  
9  
10 > fm1  
11 Nonlinear regression model  
12 model:  vel ~ Vm * conc/(K + conc)  
13 data:   puromycin  
14         K          Vm  
15 0.06412 212.68367  
16 residual sum-of-squares: 1195  
17  
18 Number of iterations to convergence: 5  
19 Achieved convergence tolerance: 4.166e-06  
20
```



Tradicionalmente es declarada la convergencia si

$$\frac{S(\boldsymbol{\theta}^{(k)}) - S(\boldsymbol{\theta}^{(k+1)})}{S(\boldsymbol{\theta}^{(k)})} < \tau_1,$$

o bien,

$$\frac{|\theta_j^{(k+1)} - \theta_j^{(k)}|}{|\theta_j^{(k+1)}|} < \tau_2,$$

donde  $\tau_1$  y  $\tau_2$  son niveles de tolerancia.

Sin embargo, más que declarar la convergencia, estos criterios corresponden a **criterios de parada** (o condiciones de término) del algoritmo.



Basado en la región de confianza del  $(1 - \alpha)\%$  para  $\theta$ , esto es,

$$\|\mathbf{f}(\theta) - \mathbf{f}(\hat{\theta})\|^2 \leq \frac{pS(\hat{\theta})}{n - p} F_{1-\alpha}(p, n - p).$$

Bates y Watts (1981)<sup>3</sup> propusieron el criterio de offset relativo,

$$\frac{\|\mathbf{e}^{(k)}\|}{\{pS(\theta^{(k)})/(n - p)\}^{1/2}} < \tau,$$

donde  $\mathbf{e}^{(k)} = \mathbf{Y} - \mathbf{f}(\theta^{(k)})$ , y  $\tau$  es un nivel de tolerancia. Este criterio se encuentra implementado en la función `nls` de R.

---

<sup>3</sup>Technometrics 23, 179-183



## Resultados de estimación

Sabemos que la función de **log-verosimilitud** está dada por

$$\ell(\psi) = -\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} S(\theta), \quad \psi = (\theta^\top, \sigma^2)^\top$$

donde la **suma de cuadrados residual** es  $S(\theta) = \|\mathbf{Y} - \mathbf{f}(\theta)\|^2$ .

Para nuestros datos, obtenemos  $S(\hat{\theta})$ ,  $\ell(\hat{\psi})$  y  $\hat{\theta}$  usando

```
1 > deviance(fm1)
2 [1] 1195.449
3
4 > logLik(fm1)
5 'log Lik.' -44.63548 (df=3)
6
7 > coef(fm1)
8      K      Vm
9 0.06412116 212.68366582
```



## Resultados de estimación

Detalles sobre la estimación pueden ser obtenidos usando la función `summary()`

```
1 > summary(fm1)
2
3 Formula: vel ~ Vm * conc / (K + conc)
4
5 Parameters:
6      Estimate Std. Error t value Pr(>|t|)
7 K    6.412e-02  8.281e-03   7.743 1.57e-05
8 Vm   2.127e+02  6.947e+00  30.615 3.24e-11
9
10 Residual standard error: 10.93 on 10 degrees of freedom
11
12 Number of iterations to convergence: 5
13 Achieved convergence tolerance: 4.166e-06
14
```



La región de verosimilitud conjunta aproximada con un nivel de confianza  $1 - \alpha$  para  $\theta$  es

$$S(\theta) \leq S(\hat{\theta}) \left\{ 1 + \frac{p}{n-p} F_{1-\alpha}(p, n-p) \right\},$$

en nuestro caso,

$$S(\theta) = \sum_{i=1}^{12} \left( \text{velocidad}_i - \frac{V_m \text{concentracion}_i}{K + \text{concentracion}_i} \right)^2$$



## Resultados de estimación

Gráfico del **modelo ajustado**:

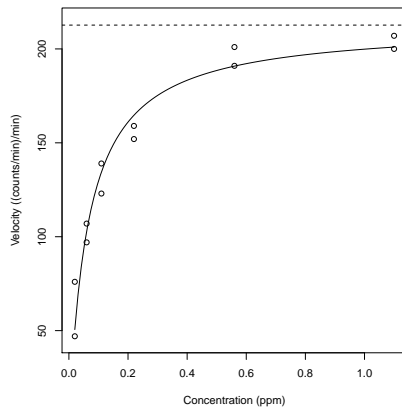
```
1 > plot(vel ~ conc, data = puromycin, ylim = c(45, 215),
2 +      xlab = "Concentration (ppm)",
3 +      ylab = "Velocity ((counts/min)/min)")
4
5 > concVal <- with(puromycin, seq(min(conc), max(conc), length.out =
6   100))
7 > lines(concVal, predict(fm1, newdata = data.frame(conc = concVal)))
8 > abline(h = coef(fm1)[2], lty = 8)
```

Gráfico del **elipsoide de confianza** del 95% para  $\theta = (V_m, K)^T$

```
1 > fm1.con <- nlsContourRSS(fm1)
2 100%
3 RSS contour surface array returned
4
5 > plot(fm1.con, nlev = 10)
```

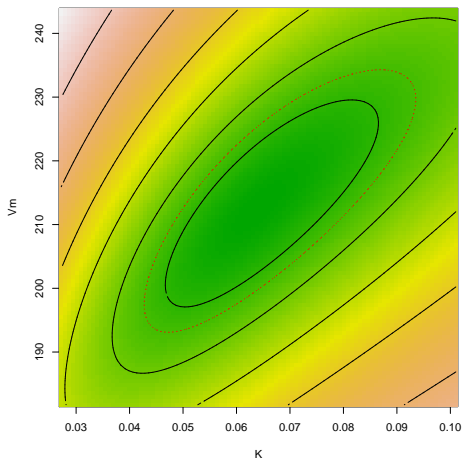


# Gráfico de ajuste para los datos de Puromycin



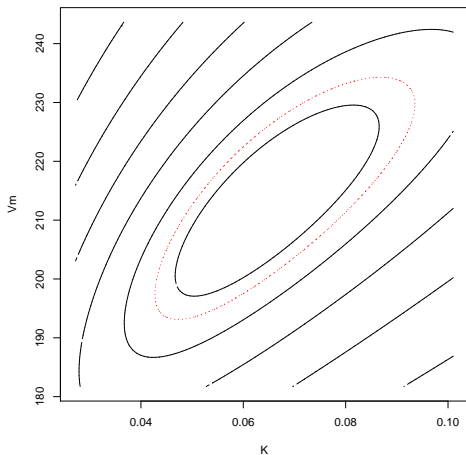
## Gráfico de contornos de la suma de cuadrados residual

La región indicada en la línea punteada representa un elipsoide de confianza del 95%.



## Gráfico de contornos de la suma de cuadrados residual

La región indicada en la línea punteada representa un elipsoide de confianza del 95%.



Bates y Watts (1988)<sup>4</sup> describen técnicas para determinar **estimaciones iniciales** en un modelo de regresión no lineal. Algunas de estas estrategias son:

- ▶ Tomar provecho de modelos parcialmente lineales, **sólo son necesarios** valores iniciales para aquellos parámetros no lineales.
- ▶ Escoger estimaciones iniciales que tengan alguna **interpretación significativa**.
- ▶ Refinar estimaciones de algunos parámetros por **iterar sobre ellos**, mientras que los otros parámetros se mantienen fijados

---

<sup>4</sup>Nonlinear Regression Analysis and Its Applications. Wiley, New York





Recuerde que, el modelo Michaelis-Menten,

$$f(x, \boldsymbol{\theta}) = \frac{V_m x}{K + x}, \quad \boldsymbol{\theta} = (V_m, K)^\top.$$

Es un modelo **intrínsecamente lineal**, pues

$$\frac{1}{f} = \frac{1}{V_m} \left( 1 + \frac{K}{x} \right) = \frac{1}{V_m} + \frac{K}{V_m} \frac{1}{x} = \beta_1 + \beta_2 z.$$

Esta estrategia es implementada en la función `SSmicmen()`.



En particular, la función `SSmicmen()`, realiza los siguientes cálculos:

```
1  # ajusta un modelo lineal
2  lin <- lm(1/vel ~ I(1/conc), data = puromycin)
3  pars <- coef(lin)
4
5  # refina estimación inicial via iteración
6  obj <- nls(y ~ x/(K + x), data = puromycin,
7  +         start = list(K = abs(pars[2]/pars[1])),
8  +         algorithm = "plinear")
9
10 # retorna 'valores iniciales'
11 start <- c(obj[2], obj[1])
12
```



La función `SSmicmen()` está documentada como parte de la ayuda de `nls`. Note que **no se requiere** que el usuario defina valores iniciales.

```
1 > fm2 <- nls(vel ~ SSmicmen(conc, Vm, K), data = puromycin, trace =  
  TRUE)  
2 1195.449 : 212.68370749 0.06412123  
3  
4 > fm2  
5 Nonlinear regression model  
6 model: vel ~ SSmicmen(conc, Vm, K)  
7 data: puromycin  
8 Vm K  
9 212.68371 0.06412  
10 residual sum-of-squares: 1195  
11  
12 Number of iterations to convergence: 0  
13 Achieved convergence tolerance: 1.917e-06
```

*Sin embargo, las funciones Self-starting están disponibles sólo para algunos tipos de funciones no lineales*



Note que, el modelo Michaelis-Menten es **condicionalmente lineal**

$$f(x, \theta) = \frac{V_m x}{K + x} = V_m \left( \frac{x}{K + x} \right),$$

podemos usar el **algoritmo de Golub-Pereyra (1973)**

```
1 > fm3 <- nls(vel ~ conc / (K + conc), data = puromycin,  
2 + algorithm = "plinear",  
3 + start = list(K = 0.08), trace = TRUE)  
4 1524.682 :    0.0800 222.2759  
5 1195.471 :    0.06423932 212.75947067  
6 1195.449 :    0.06413257 212.69098858  
7 1195.449 :    0.06412237 212.68444089  
8 1195.449 :    0.06412139 212.68381040  
9
```

Detalles de la estimación en el modelo condicionalmente lineal:

```
1 > summary(fm3)
2
3 Formula: vel ~ conc/(K + conc)
4
5 Parameters:
6      Estimate Std. Error t value Pr(>|t|)
7 K      6.412e-02  8.281e-03   7.743 1.57e-05 ***
8 .lin  2.127e+02  6.947e+00  30.615 3.24e-11 ***
9
10 Residual standard error: 10.93 on 10 degrees of freedom
11
12 Number of iterations to convergence: 4
13 Achieved convergence tolerance: 8.037e-06
14
15 > deviance(fm3)
16 [1] 1195.449
17
18 > logLik(fm3)
19 'log Lik.' -44.63548 (df=3)
```



## Usando nl2sol desde PORT

```
1 > fm4 <- nls(vel ~ Vm * conc / (K + conc), data = puromycin,
2 +   algorithm = "port",
3 +   start = list(K = 0.08), trace = TRUE)
4 0:   1577.5021: 0.0800000  205.000
5 1:   1390.1712: 0.0784593  205.959
6 2:    600.78114: 0.0632716  213.046
7 3:    597.73075: 0.0640318  212.629
8 4:    597.72447: 0.0641126  212.678
9 5:    597.72441: 0.0641204  212.683
10 6:    597.72441: 0.0641212  212.684
11 7:    597.72441: 0.0641213  212.684
12
13 > fm4
14 Nonlinear regression model
15 model: vel ~ Vm * conc/(K + conc)
16 data: puromycin
17      K      Vm
18 0.06412 212.68374
19 residual sum-of-squares: 1195
20
21 Algorithm "port", convergence message: relative convergence (4)
22
23 > deviance(fm4)
24 [1] 1195.449
25
26 > logLik(fm4)
27 'log Lik.' -44.63548 (df=3)
28
```



# Supliendo información del gradiente

Pasamos la **información del gradiente** (y la propia función  $f$ ) usando, por ejemplo:

```
1 MMgrad <- function(conc, Vm, K) {  
2   numer <- Vm * conc  
3   denom <- K + conc  
4   mean <- numer / denom  
5   partialVm <- mean / Vm  
6   partialK <- -numer / (denom^2)  
7   attr(mean, "gradient") <- cbind(partialVm, partialK)  
8   return(mean)  
9 }  
10
```



## Resumen de estimación

```
1 > fm5 <- nls(vel ~ MMgrad(conc, Vm, K), data = puromycin, trace =  
  TRUE,  
2 +   start = list(Vm = 205, K = 0.08))  
3 3155.004 : 205.00 0.08  
4 1205.662 : 213.02889449 0.06289222  
5 1195.477 : 212.60337549 0.06398775  
6 1195.449 : 212.67543430 0.06410830  
7 1195.449 : 212.68294013 0.06412003  
8 1195.449 : 212.68366575 0.06412116  
9  
10 > fm5  
11 Nonlinear regression model  
12 model: vel ~ MMgrad(conc, Vm, K)  
13 data: puromycin  
14 Vm K  
15 212.68367 0.06412  
16 residual sum-of-squares: 1195  
17  
18 Number of iterations to convergence: 5  
19 Achieved convergence tolerance: 4.161e-06
```





Resumen de estimación en el **modelo Michaelis-Menten**:

$$f(x, \boldsymbol{\theta}) = \frac{V_m x}{K + x}, \quad \boldsymbol{\theta} = (V_m, K)^\top.$$

Método	$\hat{V}_m$	$\hat{K}$	iteraciones	$S(\hat{\boldsymbol{\theta}})$
Gauss-Newton	212.68367	0.06412	5	1195.449
selfStart	212.68371	0.06412	0	1195.449
Golub-Pereira	212.68381	0.06412	4	1195.449
nl2sol	212.68374	0.06412	7	1195.449
G-N gradiente	212.68367	0.06412	5	1195.449

Se usó (según corresponda)  $\boldsymbol{\theta}^{(0)} = (205, 0.08)^\top$ .

## Funciones Self-starting disponibles

Repertorio de funciones Self-starting disponibles para **nls** (y extensiones):

**SSasymp** Modelo de regresión asintótico.

**SSasympOff** Modelo de regresión asintótico con un offset.

**SSasympOrig** Modelo de regresión asintótico a través del origen.

**SSbiexp** Modelo biexponencial.

**SSfol** Modelo de un compartimento de primer orden.

**SSfpl** Modelo logístico de cuatro parámetros.

**SSgompertz** Modelo de crecimiento de Gompertz.

**SSlogis** Modelo logístico.

**SSmicmen** Modelo Michaelis-Menten.

**SSweibull** Modelo de curva de crecimiento Weibull.

**selfStart** Constructor de modelos no lineales Self-starting.

Estas funciones han sido usadas en contextos más generales como en la función **nlme** para ajustar modelos no lineales con efectos mixtos.



A seguir revisamos algunos aspectos sobre la estimación en [modelos parcialmente lineales](#). Mayores detalles en Golub y Pereyra (1973)<sup>5</sup>

Golub y Pereyra (1973) consideran el siguiente modelo no lineal:

$$f(\mathbf{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^p \beta_j g_j(\mathbf{x}; \boldsymbol{\alpha}), \quad \boldsymbol{\alpha} \in \mathbb{R}^k, \boldsymbol{\beta} \in \mathbb{R}^p.$$

Ellos proponen [minimizar la función](#):

$$S(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \|\mathbf{Y} - \mathbf{f}(\boldsymbol{\alpha}, \boldsymbol{\beta})\|^2 = \|\mathbf{Y} - \mathbf{G}(\boldsymbol{\alpha})\boldsymbol{\beta}\|^2. \quad (2)$$

---

<sup>5</sup>SIAM Journal on Numerical Analysis **10**, 413-432



- ▶ Golub y Pereyra (1973) proponen dos algoritmos para llevar a cabo la minimización de (2).
- ▶ Una contribución importante de ese trabajo fue obtener una expresión para la derivada de una inversa generalizada (así como de una matriz de proyección).
- ▶ Simplificaremos la exposición asumiendo que  $G(\alpha)$  tiene rango columna completo (para cualquier  $\alpha$ ).
- ▶ A la fecha este algoritmo es uno de los más eficientes para resolver problemas separables (en la media).



Suponiendo que  $\alpha$  es **fijado**, podemos optimizar (2) con relación a  $\beta$ , obteniendo

$$\hat{\beta}(\alpha) = (G^\top G)^{-1} G^\top Y, \quad G = G(\alpha).$$

Así, perfilando  $S(\alpha, \beta)$  resulta,

$$\begin{aligned} S_*(\alpha) &= S(\alpha, \hat{\beta}(\alpha)) = \|Y - G(\alpha)\hat{\beta}(\alpha)\|^2 \\ &= \|Y - H(\alpha)Y\|^2 = \|\{I - H(\alpha)\}Y\|^2, \end{aligned}$$

donde

$$H(\alpha) = G(G^\top G)^{-1} G^\top, \quad G = G(\alpha).$$

Como  $H(\alpha)$  es **matriz de proyección**, tenemos que

$$S_*(\alpha) = Y^\top \{I - H(\alpha)\} Y.$$



Suponiendo que  $\alpha$  es **fijado**, podemos optimizar (2) con relación a  $\beta$ , obteniendo

$$\hat{\beta}(\alpha) = (G^\top G)^{-1} G^\top Y, \quad G = G(\alpha).$$

Así, **perfilando**  $S(\alpha, \beta)$  resulta,

$$\begin{aligned} S_*(\alpha) &= S(\alpha, \hat{\beta}(\alpha)) = \|Y - G(\alpha)\hat{\beta}(\alpha)\|^2 \\ &= \|Y - H(\alpha)Y\|^2 = \|\{I - H(\alpha)\}Y\|^2, \end{aligned}$$

donde

$$H(\alpha) = G(G^\top G)^{-1} G^\top, \quad G = G(\alpha).$$

Como  $H(\alpha)$  es **matriz de proyección**, tenemos que

$$S_*(\alpha) = Y^\top \{I - H(\alpha)\} Y.$$



Suponiendo que  $\alpha$  es **fijado**, podemos optimizar (2) con relación a  $\beta$ , obteniendo

$$\hat{\beta}(\alpha) = (G^\top G)^{-1} G^\top Y, \quad G = G(\alpha).$$

Así, **perfilando**  $S(\alpha, \beta)$  resulta,

$$\begin{aligned} S_*(\alpha) &= S(\alpha, \hat{\beta}(\alpha)) = \|Y - G(\alpha)\hat{\beta}(\alpha)\|^2 \\ &= \|Y - H(\alpha)Y\|^2 = \|\{I - H(\alpha)\}Y\|^2, \end{aligned}$$

donde

$$H(\alpha) = G(G^\top G)^{-1} G^\top, \quad G = G(\alpha).$$

Como  $H(\alpha)$  es **matriz de proyección**, tenemos que

$$S_*(\alpha) = Y^\top \{I - H(\alpha)\} Y.$$



## Modelos parcialmente lineales: Método de Golub-Pereyra

Es decir, para implementar un [algoritmo Gauss-Newton](#), asociado a la minimización de  $S_*(\alpha)$ , debemos obtener

$$\frac{\partial S_*(\alpha)}{\partial \alpha} \quad \text{o bien} \quad \frac{\partial f_*(\alpha)}{\partial \alpha^\top},$$

donde  $f_*(\alpha) = G(\alpha)\hat{\beta}(\alpha) = H(\alpha)Y$ . De este modo, debemos calcular

$$\frac{\partial H(\alpha)}{\partial \alpha^\top},$$

lo que **no es una tarea trivial**.

En efecto, el algoritmo 2 propuesto por Golub y Pereyra (1973) adopta la forma:

$$\alpha^{(k+1)} = \alpha^{(k)} - \left\{ \frac{\partial f_*(\alpha)}{\partial \alpha^\top} \right\}^{-1} f_*(\alpha) \Big|_{\alpha=\alpha^{(k)}}.$$





Para implementar el método, considere  $\mathbf{G}^- = (\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top$  una **inversa generalizada** de  $\mathbf{G}$ . Así

$$d\mathbf{H} = d\mathbf{G}\mathbf{G}^- = (d\mathbf{G})\mathbf{G}^- + \mathbf{G} d\mathbf{G}^-$$

Aprovechando la estructura de  $\mathbf{G}^-$ , tenemos<sup>6</sup>

$$\begin{aligned} d\mathbf{H} &= (d\mathbf{G})\mathbf{G}^- - \{(d\mathbf{G})\mathbf{G}^-\}^\top \mathbf{G}\mathbf{G}^- - (\mathbf{G}\mathbf{G}^-)^\top (d\mathbf{G})\mathbf{G}^- + \{(d\mathbf{G})\mathbf{G}^-\}^\top \\ &= 2(\mathbf{I} - \mathbf{H})\mathbf{G}^- d\mathbf{G}. \end{aligned}$$

Luego, **se debe vectorizar** para obtener  $D \operatorname{vec} \mathbf{G} = \partial \operatorname{vec} \mathbf{G} / \partial \boldsymbol{\alpha}^\top$ .

---

<sup>6</sup>Golub y Pereyra **solo** usan las propiedades de una inversa generalizada.

