Tema 3 - Sesión 12
Modelos no lineales

Juan Ramón González
(jrgonzalez@creal.cat)

Departamento de Matemáticas, Universidad Autónoma de Barcelona (UAB)
Centro de Investigación en Epidemiología Ambiental (CREAL)

Barcelona, Marzo-Junio de 2012

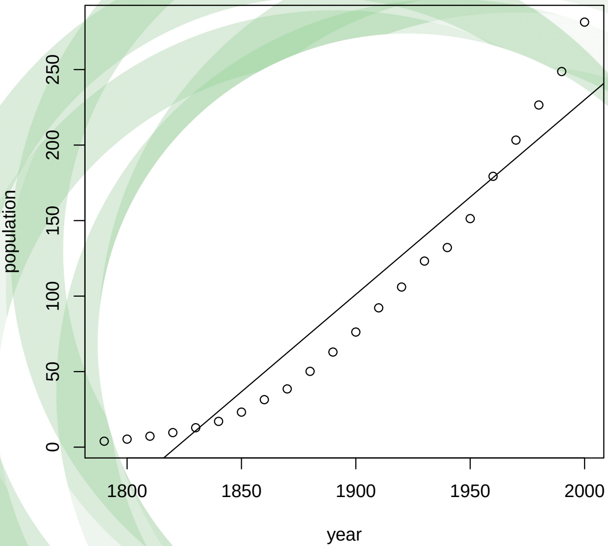
Modelos No Lineales

- Los modelos no lineales son una generalización del modelo lineal de regresión en los que la media condicionada de la variable respuesta, no es una función lineal de los parámeros
- En algunos problemas, basta con trasformar los predictores o la variable respuesta y considerar una relación lineal
- Esta aproximación es aceptable en muchas ocasiones, pero el problema radica en la interpretación de los parámetros
- Así, si el objetivo no es estimar el efecto, si no conocer aquellas variables predictoras asociadas a la variable resultado, estas transformaciones pueden ser una buena aproximación.

Modelos No Lineales

```
> library(car)
> mod <- lm(population ~ year, data=USPop)
> plot(population ~ year, data=USPop)
> abline(mod)
```

Modelos No Lineales

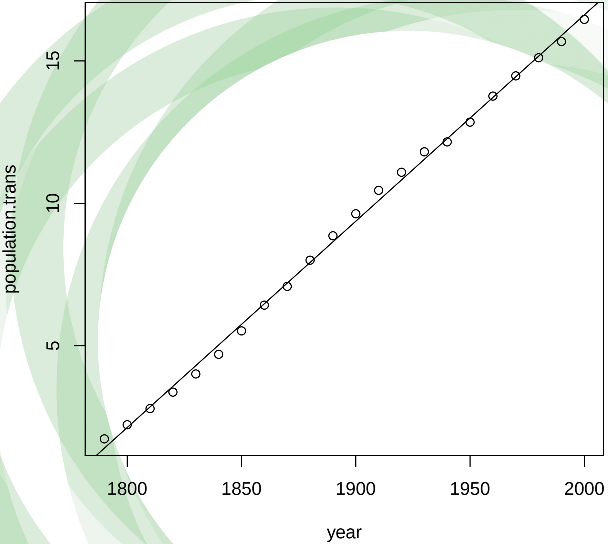


La transformación raíz cúbica puede ser adecuada

```
> trans <- powerTransform(mod)

> population.trans <- bcPower(USPop$population,
+                             coef(trans, round=TRUE))
> mod.trans <- lm(population.trans ~ year, data=USPop)
> plot(population.trans ~ year, data=USPop)
> abline(mod.trans)
```

Modelos No Lineales



Sin embargo la idea de los modelos lineales es que podemos estimar la relación

$$y = m(x, \boldsymbol{\theta}) + \epsilon$$

donde *m* puede ser cualquier función. En el caso anterior se pude utilizar el *modelo logístico de crecimiento*

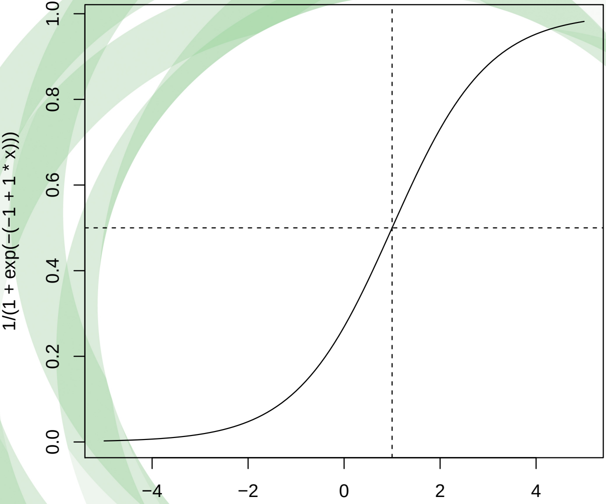
$$m(x, \boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3) = \frac{\theta_1}{1 + \exp[-(\theta_2 + \theta_3 x)]}$$

Esta función tiene el siguiente aspecto

```
> curve(1/(1+exp(-(-1 + 1*x))), from=-5, to=5, main="(b)")
> abline(h=1/2, lty=2)
> abline(v=1, lty=2)
```

Modelos No Lineales

Esta función tiene el siguiente aspecto



Modelos No Lineales

- Cambiando los parámetros $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ podemos extender o estrechar los ejes
- también se puede cambiar la velocidad a la que varía la curva entre su valor inferior y superior
- Si $\theta_3 > 0$ entonces cuando x aumenta, el término $\exp[-(\theta_2 + \theta_3 x)]$ se acerca a 0, entonces $m(x, \boldsymbol{\theta})$ se aproximará al valor θ_1 como una *asíntota* [se asume un tamaño máximo de población]
- El parámetro θ_3 controla cómo de rápida es la transición de la curva desde 0 a θ_1. Entonces, se interpreta como el parámetro de tasa de crecimiento.
- Ejemplo de uso?

Modelos No Lineales

Podemos estimar $\boldsymbol{\theta}$ minimizando la suma de los residuales al cuadrado

$$S(\boldsymbol{\theta}) = \sum w[y - m(x, \boldsymbol{\theta})]^2$$

Para ello se necesita llevar a cabo el siguiente proceso iterativo

1. Dar unos valores inciales para $\boldsymbol{\theta}$. Este paso puede ser crucial, aunque existen métodos para dar valores *razonables* [existen funciones 'self-starting' en `R`]
2. en la iteración $j \geq 1$, se da una solución t_j actualizando t_{j-1}. Si $S(t_j)$ es menor que $S(t_{j-1})$ dada una cierta cantidad (tolerancia), entonces se aumenta j en una unidad y se repite el paso anterior. Si no, entoces t_{j-1} se considera el estimador.

El algoritmo anterior debe cumplir:

1. Se necesita un método que garantice que en cada paso obtengamos un valor menor de S (o al menos que no aumente). Existen varios algoritmos para mínimos cuadrados no lineales [ver Bates and Watts (1998). Nonlinear Regression Analysis and Its Applications. Wiley, New York] Una de ellas es utilizar un algoritmo de la forma Gauss-Newton pero que en cada iteración estima las derivadas mediante métodos numéricos [métodos quasi-Newton]
2. La función S puede tener múltiples mínimos y el algoritmo podría escoger un mínimo local Una estrategia es empezar con varios puntos iniciales y ver que siempre converge a la misma solución item Puede ser que en cada iteración se mejore S y el proceso puede hacerse largo. Por ello a veces también se considera un número máximo de iteraciones como criterio para acabar el proceso de estimación. Esto podría dar problemas de encontrar mínimos locales

En R existe la función `nls` que tiene implementados estos métodos

```
> args(nls)
function (formula, data = parent.frame(), start, control = nls.control(),
    algorithm = c("default", "plinear", "port"), trace = FALSE,
    subset, weights, na.action, model = FALSE, lower = -Inf,
    upper = Inf, ...)
NULL
```

Los parámetros de control del algoritmo de minimización son:

```
> args(nls.control)
function (maxiter = 50, tol = 1e-05, minFactor = 1/1024, printEval = FALSE,
    warnOnly = FALSE)
NULL
```

Parametros iniciales

Cada problema debe tratarse de forma inidividual. Para el caso del modelo logístico de crecimiento se puede ver que:

$$y \approx \frac{\theta_1}{1 + \exp[-(\theta_2 + \theta_3 x)]} \tag{1a}$$

$$y/\theta_1 \approx \frac{1}{1 + \exp[-(\theta_2 + \theta_3 x)]} \tag{1b}$$

$$\log\left[\frac{y/\theta_1}{1 - y/\theta_1}\right] \approx \theta_2 + \theta_3 x \tag{1c}$$

De esta forma, basta con conocer un valor inicial para θ_1. Sabemos que este parámetro corresponde a la asíntota superior (máxima población en este caso). 400 parece un valor razonable teniendo en cuenta que la población estimada para 2010 era de 307 millones de habitantes).

Entonces la equación anterior se puede resolver de la siguiente forma:

```
> lm(logit(population/400) ~ year, USPop)

Call:
lm(formula = logit(population/400) ~ year, data = USPop)

Coefficients:
(Intercept)         year
  -49.24991      0.02507
```

Así nuestro vector de valores inciciales podría ser $\boldsymbol{\theta}_1 = (400, -49, 0{,}025)$

```
> mod.nl <- nls(population ~ theta1/(1 + exp(-(theta2 + theta3*year))),
+   start=list(theta1 = 400, theta2 = -49, theta3 = 0.025),
+   data=USPop, trace=TRUE)

3060.786 :  400.000 -49.000   0.025
558.5357 :  426.06199142 -42.30785623   0.02142146
457.9746 :  438.41471526 -42.83690081   0.02167713
457.8071 :  440.89027810 -42.69866517   0.02160152
457.8056 :  440.81680958 -42.70804961   0.02160649
457.8056 :  440.83444805 -42.70688446   0.02160586
457.8056 :  440.83332801 -42.70697788   0.02160591
```

```
>  summary(mod.nl)

Formula: population ~ theta1/(1 + exp(-(theta2 + theta3 * year)))

Parameters:
         Estimate Std. Error t value Pr(>|t|)
theta1 440.833328  35.000136   12.60 1.14e-10 ***
theta2 -42.706978   1.839138  -23.22 2.08e-15 ***
theta3   0.021606   0.001007   21.45 8.87e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.909 on 19 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.481e-06
```

En este tipo de modelos hay una medida que es muy importante. Esta medida es en qué punto de la variable `x` se encuentra la mitad de la asíntota de la variable `y` (la mitad del máximo que puede tomar. En este caso, en qué año observamos la mitad de la población.

En los estudios dosis-respuesta con fármacos, esta medida se conoce como *la dosis mediana* y nos da el valor de dosis para que la mitad de los individuos fallecen. En otros estudios también se conoce como valor IC_{50}.

Este valor se estima como $-\hat{\theta}_3/\hat{\theta}_2$, que en nuestro caso toma el valor

```
> -coef(mod.nl)[3]/coef(mod.nl)[2]

      theta3
0.0005059105
```

El error estandar para este ic50 se puede estimar mediante el método delta. NOTA: para el caso univariante $\mathrm{Var}(g(\theta_1)) = \sigma^2_{\theta_1} g'(\theta_1)^2$

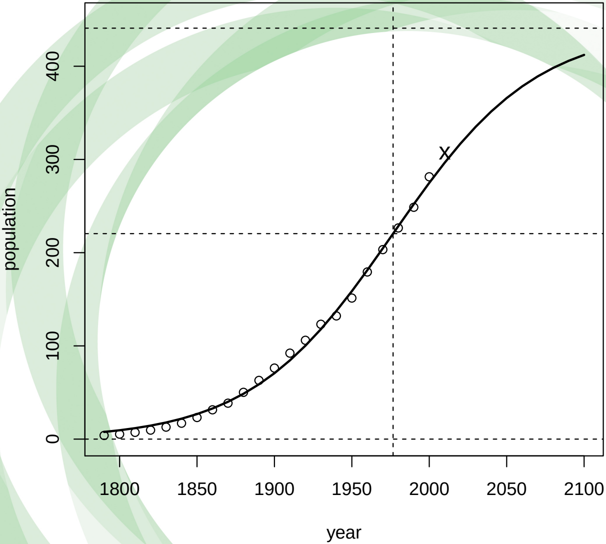
```
> deltaMethod(mod.nl, "-theta2/theta3")

             Estimate       SE
-theta2/theta3 1976.634 7.555785
```

Podemos comprobar que el modelo es adecuado para realizar predicciones

```
> plot(population ~ year, USPop, xlim=c(1790, 2100), ylim=c(0,450))
> with(USPop, lines(seq(1790, 2100, by=10),
+  predict(mod.nl, data.frame(year=seq(1790, 2100, by=10))), lwd=2))
> points(2010, 307, pch="x", cex=1.3)
> abline(h=0, lty=2)
> abline(h=coef(mod.nl)[1], lty=2)
> abline(h=.5*coef(mod.nl)[1], lty=2)
> abline(v= -coef(mod.nl)[2]/coef(mod.nl)[3], lty=2)
```

Modelos No Lineales



Self-Starting values (Pinheiro, J. C. and Bates, D. M. (2000). Mixed-E ects Models in S and S-PLUS. Springer, New York)

```
> mod.ss <- nls(population ~ SSlogis(year, phi1, phi2, phi3), data=USP
> summary(mod.ss)

Formula: population ~ SSlogis(year, phi1, phi2, phi3)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
phi1  440.834     35.000   12.60 1.14e-10 ***
phi2 1976.634      7.556  261.61  < 2e-16 ***
phi3   46.284      2.157   21.45 8.87e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.909 on 19 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.822e-06
```

Self-Starting values (Pinheiro, J. C. and Bates, D. M. (2000). Mixed-E ects Models in S and S-PLUS. Springer, New York)

El problema es que hay que leer cómo están parametrizados los modelos. En este caso como $-\hat{\theta}_3/\hat{\theta}_2$ es una medida interesante, la función se parametriza como $\phi_1 = \theta_1$, $\phi_2 = -\theta_2/\theta_3$, $\phi_3 = 1/\theta_3$, por lo que tenemos

$$m(x, \boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)) = \frac{\phi_1}{1 + \exp[-(x - \phi_2)/\phi_3)]}$$

Comprobamos que ambas parametrizaciones y el método delta dan los mismos resultados

```
> summary(mod.ss)

Formula: population ~ SSlogis(year, phi1, phi2, phi3)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
phi1  440.834     35.000   12.60 1.14e-10 ***
phi2 1976.634      7.556  261.61  < 2e-16 ***
phi3   46.284      2.157   21.45 8.87e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.909 on 19 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.822e-06

> deltaMethod(mod.nl, "1/theta3")

         Estimate       SE
1/theta3 46.28363 2.157445
```

Estos son los modelos que están implementados en `R`

Function	Equation, $m(x, \phi) =$
SSasymp	Asymptotic regression
	$\phi_1 + (\phi_2 - \phi_1)\exp[-\exp(\phi_3)x]$
SSasympOff	Asymptotic regression with an offset
	$\phi_1\{1 - \exp[-\exp(\phi_2) \times (x - \phi_3)]\}$
SSasympOrig	Asymptotic regression through the origin
	$\phi_1\{1 - \exp[-\exp(\phi_2)x]\}$
SSbiexp	Biexponential model
	$\phi_1 \exp[-\exp(\phi_2)x] + \phi_3 \exp[-\exp(\phi_4)x]$
SSfol	First-order compartment model
	$\frac{D\exp(\phi_1+\phi_2)}{\exp(\phi_3)[\exp(\phi_2)-\exp(\phi_1)]}\{\exp[-\exp(\phi_1)x] - \exp([-\exp(\phi_2)x]\}$
SSfpl	Four-parameter logistic growth model
	$\phi_1 + \frac{\phi_2-\phi_2}{1+\exp[(\phi_3-x)/\phi_4]}$
SSgompertz	Gompertz model
	$\phi_1 \exp(\phi_2 x^{\phi_3})$
SSlogis	Logistic model
	$\phi_1/(1 + \exp[(\phi_2 - x)/\phi_3])$
SSmicmen	Michaelis-Menten model
	$\phi_1 x/(\phi_2 + x)$
SSweibull	Weibull model
	$\phi_1 + (\phi_2 - \phi_1)\exp[-\exp(\phi_3)x^{\phi_4}]$

Modelos con Covariables

Muchas veces queremos estimar un modelo lineal con la misma función para distintos grupos de datos. Por ejemplo podemos comparar la población Canadiense y la de U.S.

```
> datos <- data.frame(rbind(data.frame(country="US", USPop[,1:2]),
+                     data.frame(country="Canada", CanPop)))
> some(datos)
```

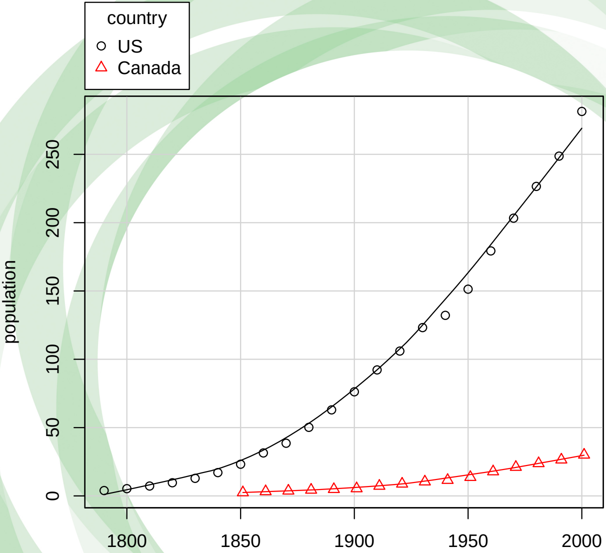
```
    country year population
3        US 1810   7.239881
4        US 1820   9.638453
6        US 1840  17.063353
7        US 1850  23.191876
8        US 1860  31.443321
12       US 1900  76.212168
18       US 1960 179.323175
20       US 1980 226.542199
71   Canada 1911   7.207000
121  Canada 1961  17.780000
```

Modelos con Covariables

Podemos visualizar los datos de la siguiente forma (usando la librería car). Poniendo box y reg igual a FALSE eliminamos los boxplots y la linea de regresión. Las lineas que se observan son suavizados no paramétricos.

```
> scatterplot(population ~ year|country, data=datos,
+       box=FALSE,  reg=FALSE)
```

Modelos con Covariables



Modelos con Covariables

Podemos estimar un modelo logístico de crecimiento de forma separada para cada grupo usando la librería `nlme`. La función `nlsList` asume la misma varianza para los errores en todos los grupos, pero para nuestro caso no es apropiado ya que la variabilidad en U.S es mayor que en Canada. Para forzar varianzas distintas, usamos el argumento `pool`

```
> library(nlme)
> mod.list <- nlsList(population ~ SSlogis(year, phi1, phi2, phi3)|cou
+                           data=datos, pool=FALSE)
> summary(mod.list)
Call:
  Model: population ~ SSlogis(year, phi1, phi2, phi3) | country
   Data: datos

Coefficients:
   phi1
        Estimate Std. Error   t value    Pr(>|t|)
US     440.83357   35.00023 12.595163 1.13903e-10
Canada  71.44637   14.15008  5.049186 2.22768e-04
   phi2
       Estimate Std. Error  t value     Pr(>|t|)
US     1976.634   7.555803 261.6048 2.942066e-35
Canada 2015.663  16.474723 122.3488 2.730058e-21
   phi3
```

Podemos usar la función `deltaMethod` para calcular el error estándard de la diferencia de la tasa de crecimiento entre ambos países. Para ello tenemos en cuenta que el objeto `mod.list` es una lista de objetos de clase `nls`.

Obtenemos los coeficientes

```
> phis <- unlist(lapply(mod.list, coef))
> phis
```

```
   US.phi1     US.phi2     US.phi3 Canada.phi1 Canada.phi2 Canada.phi3
 440.83357  1976.63417    46.28366    71.44637  2015.66308    47.74810
```

Y sus varianzas-covarianzas

```
> vars <- lapply(mod.list, vcov)
> vars
$US
           phi1      phi2      phi3
phi1 1225.01592 262.85502 69.128450
phi2  262.85502  57.09016 15.228746
phi3   69.12845  15.22875  4.654582

$Canada
          phi1      phi2      phi3
phi1 200.22464 232.47918 40.834988
phi2 232.47918 271.41651 48.460719
phi3  40.83499  48.46072  9.364042
```

Creamos la matriz de varianzas-covarianzas

```
> zero <- matrix(0, nrow=3, ncol=3)
> var <- rbind( cbind(vars[[1]], zero), cbind(zero, vars[[2]]))
> var
```

```
          phi1      phi2      phi3
phi1 1225.01592 262.85502 69.128450   0.00000   0.00000  0.000000
phi2  262.85502  57.09016 15.228746   0.00000   0.00000  0.000000
phi3   69.12845  15.22875  4.654582   0.00000   0.00000  0.000000
phi1    0.00000   0.00000  0.000000 200.22464 232.47918 40.834988
phi2    0.00000   0.00000  0.000000 232.47918 271.41651 48.460719
phi3    0.00000   0.00000  0.000000  40.83499  48.46072  9.364042
```

Calculamos la diferencia y su error estandard

```
> deltaMethod(phis, "US.phi3 - Canada.phi3", vcov=var)

                       Estimate       SE
US.phi3 - Canada.phi3 -1.464439 3.744145

> deltaMethod(phis, "US.phi2 - Canada.phi2", vcov=var)

                       Estimate       SE
US.phi2 - Canada.phi2 -39.02892 18.12475
```