Non lineal models

Juan R Gonzalez

Contents

1 Introduction

Objectives

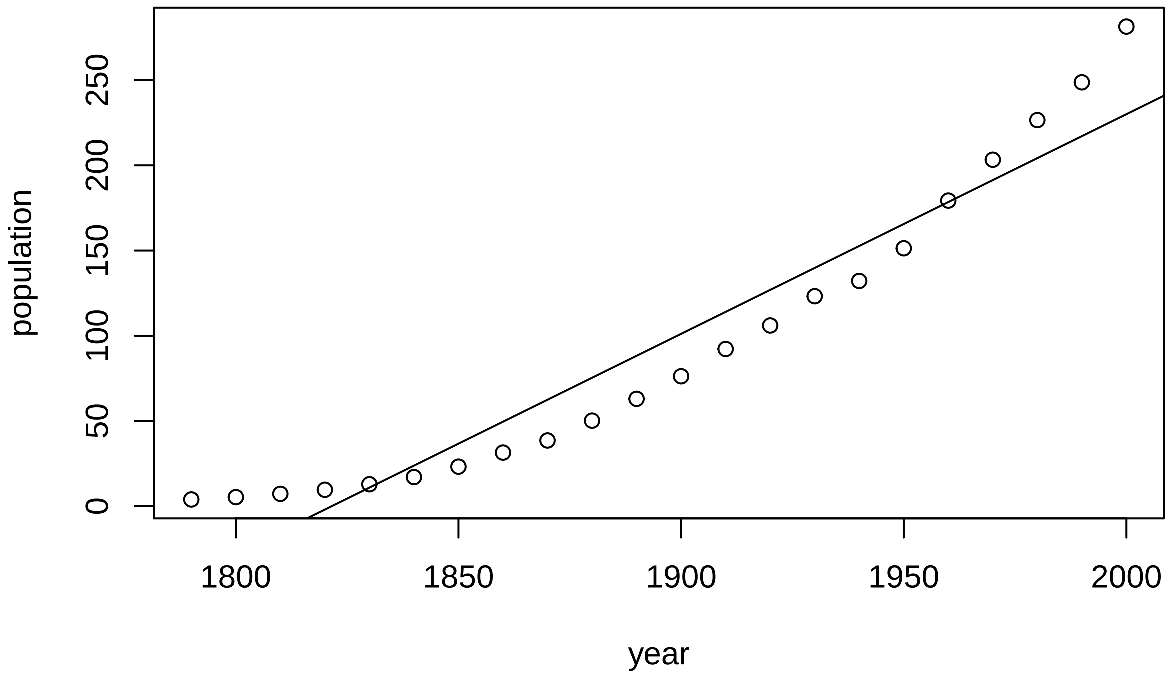
- Learn how to model longitudinal data using non-lineal models.
- Peform data analyses where the scientific question is to determine the non lineal association between longitudinal data and a continuous outcome.

2 Introduction

- Non lineal models generalize linear regression models when the mean conditioned to the response variable is a non lineal function on the parameters. 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
- In some situations, it is enough to transform either the outcome or the predictors and use lineal models
- This approach works in many situation. However, the problem arises when interpreting model parameters.
- Therefore, if our aim is to determine those variable that are associated with the outcome, this approach is a good option. On the other hand, is our aim is to interpret model parameters, non lineal methods have to be used instead.

Let us illustrate this situation with a real data example. Next figure depicts the linear relationship between US population and different years (i.e. longitudinal data)

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



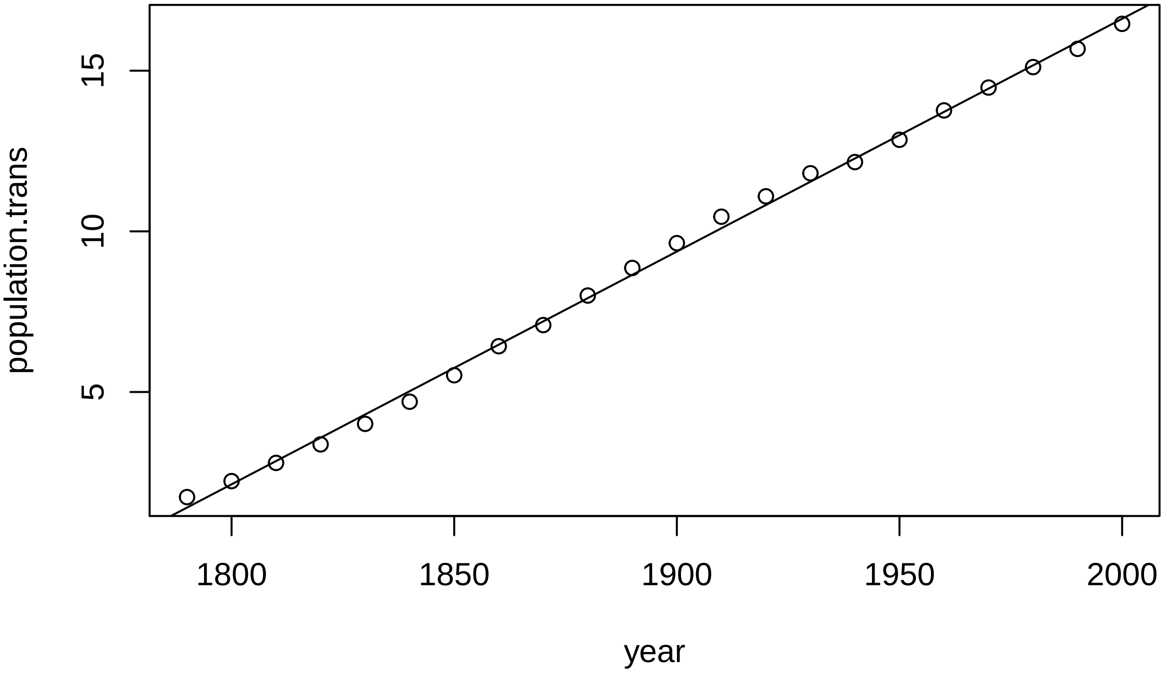
We clearly observe that assuming a linear relationship is not fitting properlly our data. One option may be to look for a transformation that guarantees linear association. This can be perform as following:

```
trans <- powerTransform(mod)
trans
## Estimated transformation parameters
##       Y1
## 0.344935
```

The function `powerTransform` from car packages indicates that cubic root may be a good transformation ($3 \sim 1/0.3449$). Then, the linear analysis can be performed by using this transformation by executing:

```
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)
```



Now the lineal relationship between the transformed variable and the year is very clear. However, model parameters are hard to be interpreted. The idea behing the non lineal models is that, when the non linear relatioship is known, model parameters can be estimated and, hence, interpretation can be facilitated. In general, we can estimate the relationship

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

where m can be any function. In our example, we can use the *growth logistic model* that can be expressed as:

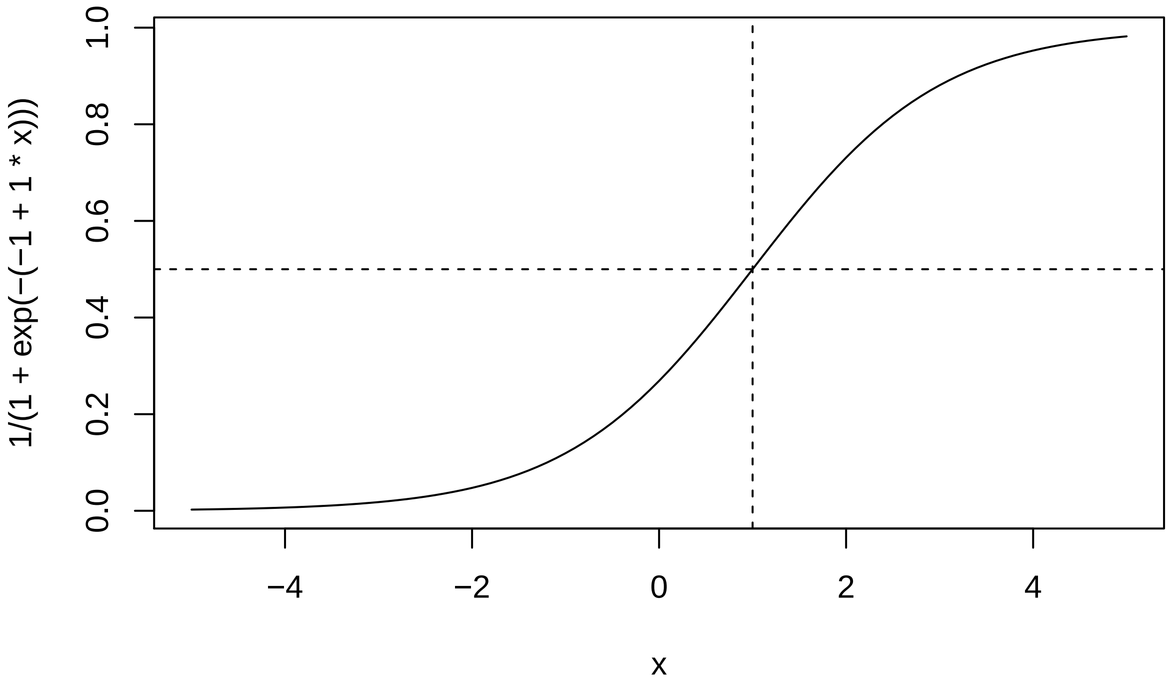
$$m(x, \boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3) = \frac{\theta_1}{1 + \exp[-(\theta_2 + \theta_3 x)]}$$

Here,

- Changing the parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ we can set axes limits
- The velocity how curve is varying between lower and upper limits can also be set. For instance, if $\theta_3 > 0$ then when x increases the term $\exp[-(\theta_2 + \theta_3 x)]$ tends to 0. Therefore $m(x, \boldsymbol{\theta})$ will approximate to θ_1 as its asymptote. In other words, we assume a maximum population size
- The parameter θ_3 controls how fast is the transition of the curve between 0 and θ_1. This parameter is known as the growth rate.

This plot shows the expected curve when all parameters are set to 1.

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



3 Model parameter estimates

We can estimate $\boldsymbol{\theta}$ minimizing the sum of the square residuals

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

To this end, the next iterative process must be performed:

- Provide intial values of $\boldsymbol{\theta}$. This step can be crucial. There are methods to provide *reasonable* start values. There are *self-starting* functions in `R` to do this task.
- The iteration $j \geq 1$ gives a solution t_j updating t_{j-1}. If $S(t_j)$ is lower thatn $S(t_{j-1})$ given a tolerance, then j is augmented in one unity and the previous step is repeated. If not, the solution in t_{j-1} is considered the estimator.

This algorith must hold:

- The method has to provide a lower value of S at each step. There are several algorithms to perform this (see Bates and Watts). One of them is to use an algorithm based on Gauss-Newton method estimating derivatives at each step using numerical methods (e.g. quasi-Newton).
- The S function can have several solutions and the algorithm may get a local minima. One strategy to avoid this problem can be to start by using several initial points and check whehter they converge to the same answer.
- In some ocassions the process may provide solutions that improve S and the process may be long. This can be overcame by fixing a maximum number of iterations. This may provide a local minima and should be carefully checked.

In `R`there exists a function called `nls` that has implemented these methods.

```
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
```

And the parameters to control the algorithm (e.g. maximum number of iterations, tolerance, ...) are:

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

The initial value of each problem must be considered independently. In our case, in the logistic model it can be seen that:

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

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

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

In this case, it is enough to know the intial value of θ_1. We know that this parameter corresponds to the upper asymptote (maximum number of individuals in the population in our example). 400 seems to ve a reasonable value taking into account that the estimated pupulation in 2010 was 307 million inhabitants. The previous equation can be solved by:

```
lm(logit(population/400) ~ year, USPop)
##
## Call:
## lm(formula = logit(population/400) ~ year, data = USPop)
##
## Coefficients:
## (Intercept)         year
##   -49.24991      0.02507
```

Therefore, our vector of initial values (start argument) could be $\boldsymbol{\theta_1} = (400, -49, 0.025)$. The non lineal model is fitted in R by:

```
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 :  438.41469905 -42.83690177   0.02167713
## 457.8071 :  440.89033603 -42.69866176   0.02160152
## 457.8056 :  440.81680693 -42.70804988   0.02160649
## 457.8056 :  440.83447052 -42.70688318   0.02160586
## 457.8056 :  440.83334419 -42.70697695   0.02160591
summary(mod.nl)
##
## Formula: population ~ theta1/(1 + exp(-(theta2 + theta3 * year)))
##
## Parameters:
##          Estimate Std. Error t value Pr(>|t|)
## theta1 440.833344  35.000138   12.60 1.14e-10 ***
## theta2 -42.706977   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: 1.239e-06
```

In these type of models there is a very relevant measurement. We are normally interested in knowing where the mean value of the asymptote of variable *y* is located. In our example, this corresponds to the year where the half of the maximum population will be observed.

In the dosage-response studies where the effect of a drug is studied, this value is know as the *median dosage* and it provides the value of the dosage where half of the inviduals die. This is know as IC_{50} value.

It can be estimated as $-\hat{\theta}_3/\hat{\theta}_2$. In our case

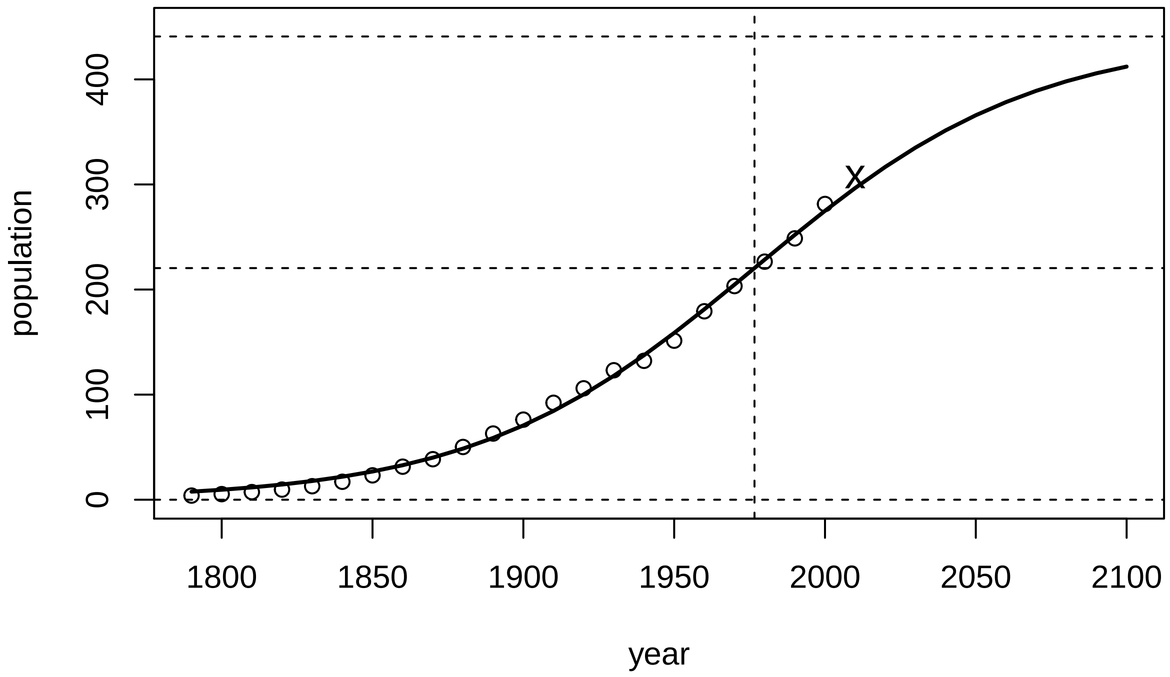
```
-coef(mod.nl)[3]/coef(mod.nl)[2]
##       theta3
## 0.0005059105
```

The standard error of IC50 can be estimated by means of the delta method. NOTE: for teh univariate case this corresponds to Var$(g(\theta_1)) = \sigma^2_{\theta_1} g'(\theta_1)^2$. However this cannot be applied in our case since IC50 is computed by using θ_2 and θ_3. In `R` the delta method can be computed by

```
deltaMethod(mod.nl, "-theta2/theta3")
##                Estimate       SE    2.5 %   97.5 %
## -theta2/theta3 1976.634 7.555785 1961.825 1991.443
```

We can verify that this model is correct to make predictions

```
 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)
```



In R some of the most commonly used non lineal functions are already implemented. This is the case of logistic growth model that is implemented in the function `SSlogis` as described Pinheiro and Bates. This function can be used in our example as following:

```
mod.ss <- nls(population ~ SSlogis(year, phi1, phi2, phi3), data=USPop)
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: 3.826e-06
```

The problem is that we must know how model is parameterized. In this case as $-\hat{\theta}_3/\hat{\theta}_2$ is an important measurement, the function is parameterized by $\phi_1 = \theta_1$, $\phi_2 = -\theta_2/\theta_3$, $\phi_3 = 1/\theta_3$. We have

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

We can verify as both methods are providing the same solution

```
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: 3.826e-06
deltaMethod(mod.nl, "1/theta3")
##          Estimate       SE    2.5 %   97.5 %
## 1/theta3 46.28363 2.157445 42.05512 50.51215
```

These are the non linear models that are implemented in `R`

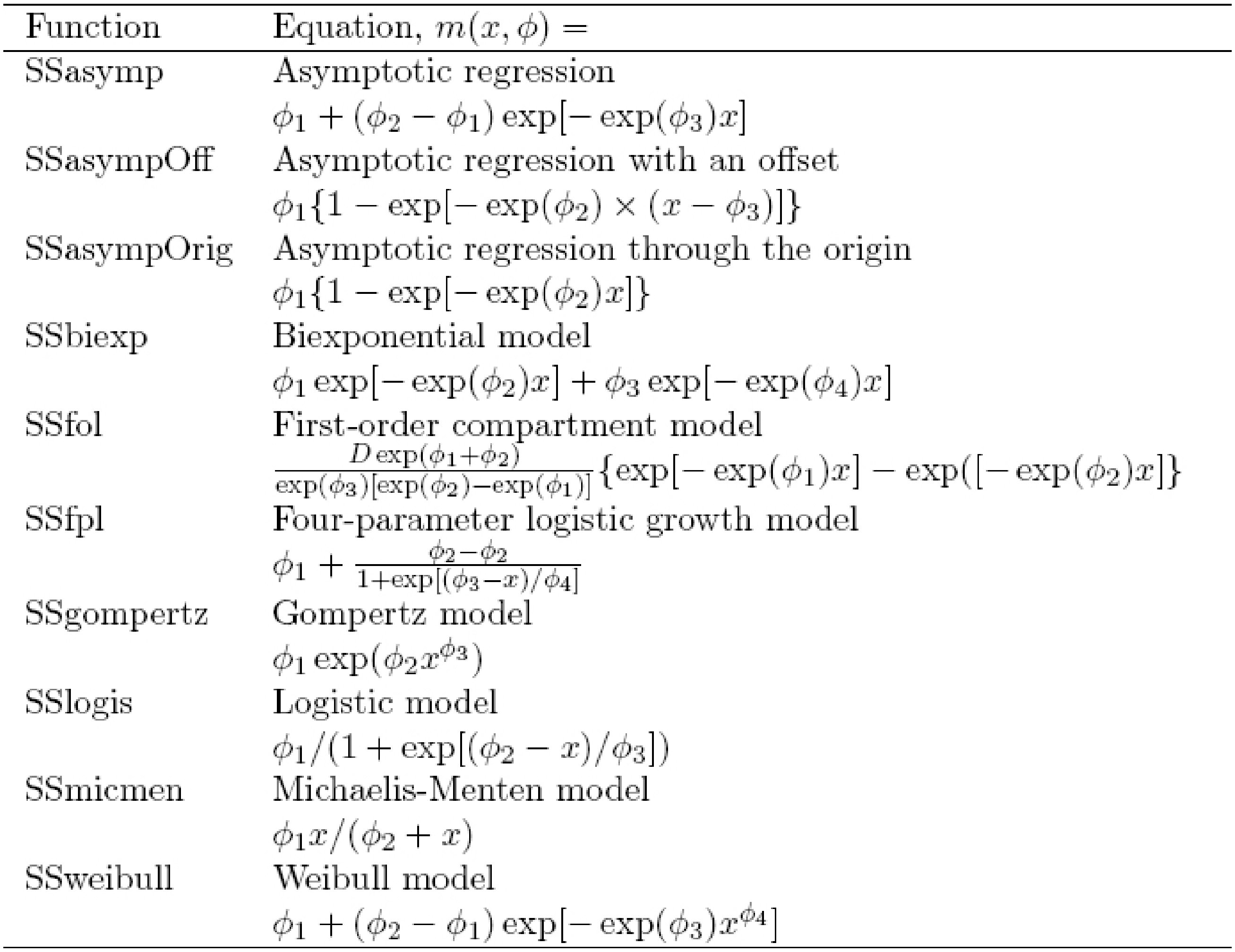
4 Model with covariates

In many occassions we are interested in estimating a non lineal model with the same function for different groups of data. For instance, one may be interested in comparing population from Canada and US.

```
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
## 5        US 1830  12.860702
## 9        US 1870  38.558371
## 17       US 1950 151.325798
## 19       US 1970 203.302031
## 110  Canada 1851   2.436000
## 71   Canada 1911   7.207000
## 101  Canada 1941  11.507000
## 111  Canada 1951  13.648000
## 141  Canada 1981  23.774000
```

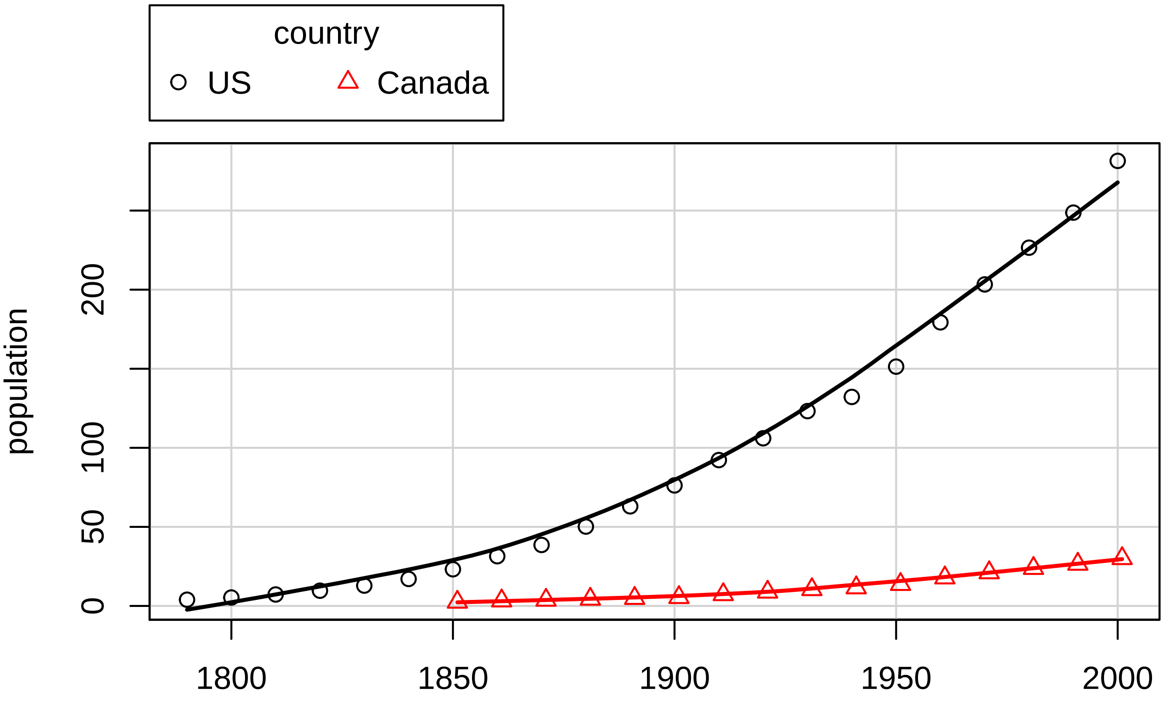
We can visualize the data as following (NOTE: lines are estimated using non parametric methods)

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



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
	$\phi_1 + (\phi_2 - \phi_1)\exp[-\exp(\phi_3)x^{\phi_4}]$

Figure 1: alt text



We can estimate a logistic growth model separately of each group by using the library `nlme`. The function `nlsList` assumes the same variance of the errors in all groups. However, in our example, the variability observed in US is larger than in Canada. To force different variances, the argument `pool` must be set to `FALSE`.

```
library(nlme)
mod.list <- nlsList(population ~ SSlogis(year, phi1, phi2, phi3)|country,
                     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 1.139030e-10
## Canada  71.44636   14.15007  5.049186 2.227679e-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 2.730057e-21
##    phi3
##        Estimate Std. Error  t value     Pr(>|t|)
## US     46.28366   2.157448 21.45297 8.867045e-15
## Canada 47.74810   3.060072 15.60359 8.477325e-10
```

We can use the function `deltaMethod` to compute the standard error of the difference between the growth rate between both countries. To this end, we must consider that the object `mod.list` is a list of objects of class `nls`:

```
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    71.44636  2015.66308    47.74810
```

Their variances and covariances can be obtained by:

```
vars <- lapply(mod.list, vcov)
vars
## $US
##             phi1      phi2      phi3
## phi1 1225.01594 262.85502 69.128451
## phi2  262.85502  57.09016 15.228746
## phi3   69.12845  15.22875  4.654582
##
## $Canada
##            phi1      phi2      phi3
## phi1 200.22461 232.47915 40.834985
## phi2 232.47915 271.41649 48.460716
## phi3  40.83498  48.46072  9.364042
```

We can create the matrix of variances and covariances like this:

```
zero <- matrix(0, nrow=3, ncol=3)
var <- rbind( cbind(vars[[1]], zero), cbind(zero, vars[[2]]))
var
##             phi1      phi2      phi3
## phi1 1225.01594 262.85502 69.128451   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 200.22461 232.47915 40.834985
## phi2    0.00000   0.00000  0.000000 232.47915 271.41649 48.460716
## phi3    0.00000   0.00000  0.000000  40.83498  48.46072  9.364042
```

And we can perform a formal comparison of any parameter by:

```
deltaMethod(phis, "US.phi3 - Canada.phi3", vcov=var)
##                        Estimate       SE     2.5 %   97.5 %
## US.phi3 - Canada.phi3 -1.464439 3.744145 -8.802829 5.873951
deltaMethod(phis, "US.phi2 - Canada.phi2", vcov=var)
##                        Estimate       SE     2.5 %    97.5 %
## US.phi2 - Canada.phi2 -39.02892 18.12475 -74.55278 -3.505054
```

5 Exercise (to deliver)

Exercise 1:

In biochemistry, the kinetic model of Michaelis-Menten is used to analyzed enzyme kinetics. This model relates the rate reaction v with the sustrate concentration S by means of the next equation:

$$v = \frac{\phi_1}{\phi_2 + S}$$

where ϕ_1 corresponds to the maximum reaction rate achieved by the system (saturating value) and ϕ_2 (known as Michaelis constant) corresponds to the concentration where reaction rate is half of ϕ_1. This parameter ϕ_2 is highly relevant to the biologists.

- Draw the theoretical curve then $\phi_1 = 3.5$ y $\phi_2 = 0.4$ of a range of contentration values between 0 and 5.
- The file `kinetics.txt` contains information of a experiment carried out to estimate the concentration at which the reaction rate is half of it maximum. Use the Michaelis-Menten model to estimate this parameter and its confidence interval (CI) at 95% (NOTE: investigate whether there is any generic `R` function to automatically compute the CI).

Exercise 2:

The file `ic50.txt` contains information about cellular growth accross time (variable *tiempo*) of three different exposures (variables *low*, *medium*, *high*).

- Create a plot comparin the three growth curves
- Calculate the *ic50* (time at which half of the maximum cellular grotwh is achieved) for the three types of exposure by using the model you think that best approximate the data. Is there statistically significant differences between those values?
- Create a plot with observed and predicted values and verify whether the model you have used fit the data.
- Which type of exposure has the larger growth rate?

6 References

- The [`nlme`] package (https://cran.r-project.org/web/packages/nlme/)
- Bates and Watts (1998). Nonlinear Regression Analysis and Its Applications. Wiley, New York.

- Pinheiro, J. C. and Bates, D. M. (2000). Mixed-effects Models in S and S-PLUS. Springer, New York.

7 Session information

```
## R version 3.3.2 (2016-10-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 14393)
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.1252  LC_CTYPE=Spanish_Spain.1252
## [3] LC_MONETARY=Spanish_Spain.1252 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] nlme_3.1-128    car_2.1-4       knitr_1.15.1    BiocStyle_2.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.9        magrittr_1.5       splines_3.3.2
##  [4] MASS_7.3-45        lattice_0.20-34    minqa_1.2.4
##  [7] stringr_1.2.0      tools_3.3.2        nnet_7.3-12
## [10] pbkrtest_0.4-6     parallel_3.3.2     grid_3.3.2
## [13] mgcv_1.8-15        quantreg_5.29      MatrixModels_0.4-1
## [16] htmltools_0.3.5    yaml_2.1.14        lme4_1.1-12
## [19] rprojroot_1.2      digest_0.6.11      Matrix_1.2-7.1
## [22] nloptr_1.0.4       evaluate_0.10      rmarkdown_1.3
## [25] stringi_1.1.2      backports_1.0.5    SparseM_1.74
```