Non-linear regression - Why you shouldn't take the logarithms of your variables

Koen Van de moortel, MSc experimental physics
Jules de Saint-Genoisstraat 98
9050 Gentbrugge, Belgium
info@lerenisplezant.be
15 Feb. 2021, updated 17 May 2021

Abstract

'Preprocessing' data by taking the logarithm of the variables is a bad practice. This is illustrated here with examples.

DOI: 10.13140/RG.2.2.18442.80324 (first version)

Keywords: regression, least squares, non-linear, iteration, logarithms, data analysis, graphmatica, geogebra, TI-84, curve fitting, mathematical modeling.

Introduction

Many popular regression software programs like the TI-84 calculator, Graphmatica, GeoGebra, and even Wolfram, known for its sublime math software[1], etc. make a mistake in their algorithms for non-linear regression and yet, very few people seem to notice this!

What happens? The algorithm takes the logarithms of the variables and then the formulas for linear regression can be used. For example:

$$y = a \cdot x^b \Rightarrow \log(y) = b \cdot \log(x) + \log(a)$$

At first sight this looks perfectly okay: there is a linear relationship between log(x) en log(y), definitely yes. And for linear regression a simple straightforward algorithm exists to find the best fitting a and b for a given dataset with values (x_i, y_i, i=1..n), while exponential regression requires 'time consuming' iteration. Textbooks recomment to use it [e.g. Dukkipati 2010, p. 207]. So what's the problem?

Problem 1: weights

It's very simple: suppose the precision is the same for all the measurements, this will not be the case anymore for their logarithms! Each measurement obtains a different weight. (The problem is mentioned in 'The Engineering Statistics Handbook' [2].)

Example: if y_1 = 1000 ± 10 en y_2 = 100 ± 10, then $\log(y_1)$ = 3 ± 0.0043 and $\log(y_2)$ = 2 ± 0.044. Smaller y values will therefore have a much larger leverage! [3]

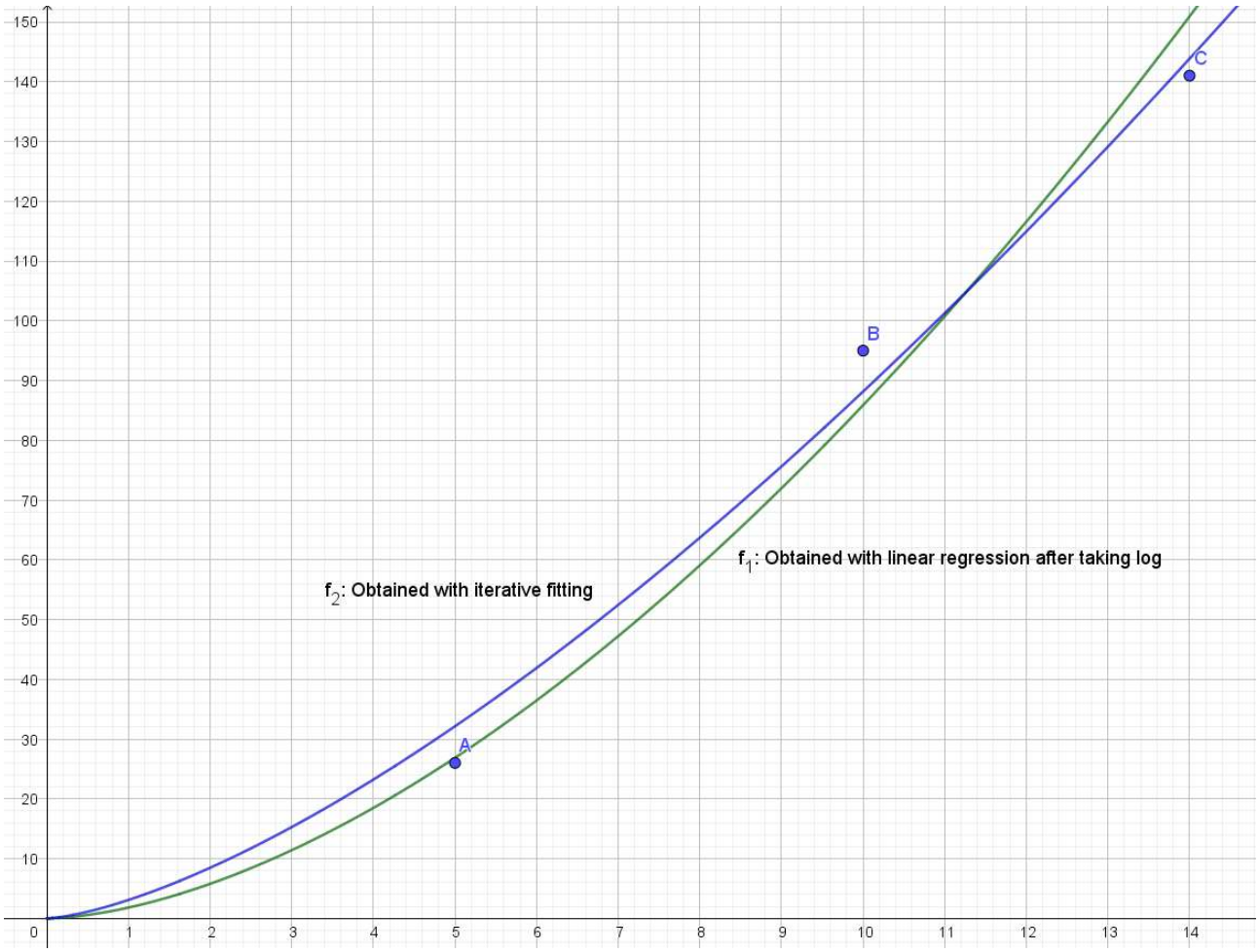
Let's invent some data and feed them to the algorithms:

point	x	y	log(x)	log(y)
A	5	26	0.699	1.415
B	10	95	1	1.978
C	14	141	1.146	2.149

The TI-84 calculator and GeoGebra produce: a = 1.8106 and b = 1.6760.

My own software, 'FittingKVdm' (version 1: April 2021), approaches the parameters iteratively, minimizing step by step the sum of the (weighted) squares of the differences between the measured and the calculated y values ('chi square per degree of freedom' = χ^2). In this example the weights are assumed to be the same, equal to 1.
The result: a = 3.0896 and b = 1.4553.

The difference is quite significant, and clearly visible in the graph below: with the first parameter values (f_1), the point A has more leverage; A is very close to the curve, but B and C are relatively far; χ^2 = 13.509. With the second parameter set (f_2), the curve moves nicely between all points, and χ^2 = 9.632, so this is objectively a better fit.



Problem 2: zeroes

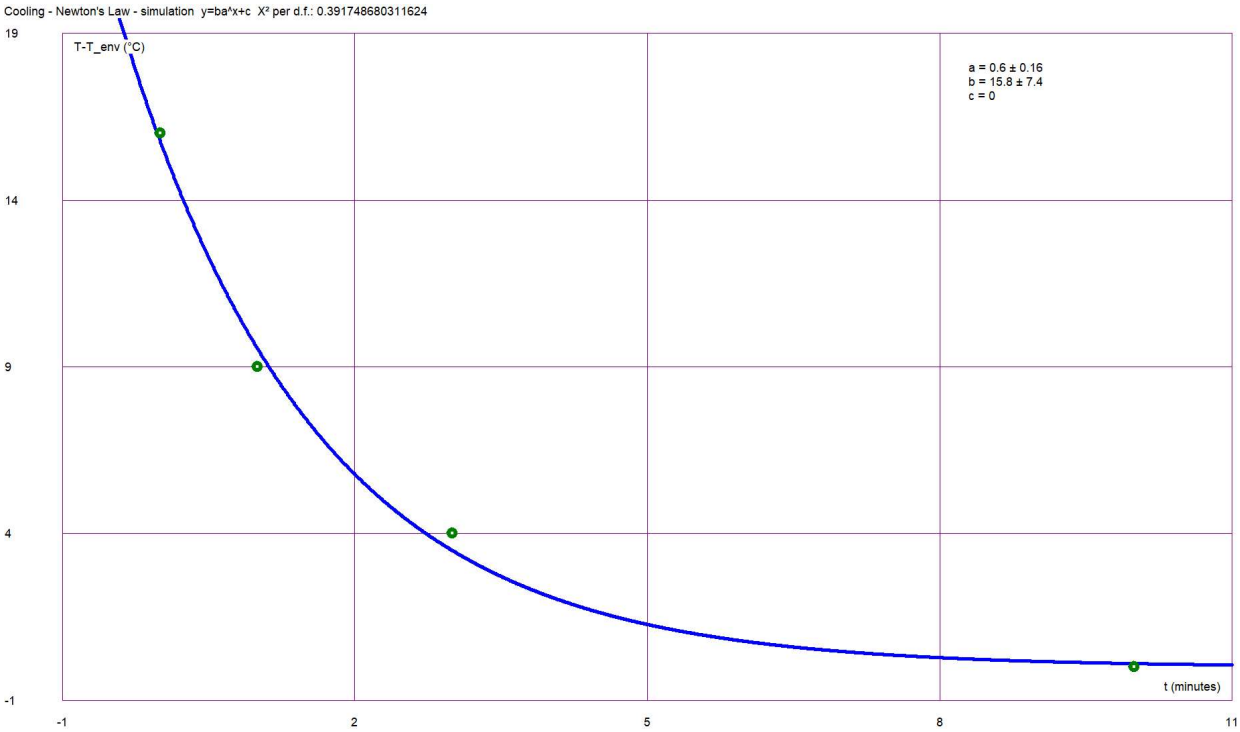
Suppose you want to test the thermal insulation of a recipient. You put some hot water in it and measure the temperature in excess of the environment (let's call this $y=T-T_{env}$, in °C) at different times (t in minutes). According to Newton's cooling law, y will decrease exponentially to zero.

Suppose we have measured these data, using a thermometer with a resolution of 1°C:

t	0	1	3	10
y	16	9	4	0

Now let's try and fit an exponential curve through these data: $y = b \cdot a^t$.
Ideally, a would be 1 (perfect insulation), and the worst case is a=0. The parameter b should be the starting temperature excess, so around 16 in this case.

FittingKVdm 1.0 finds: a = 0.60487 and b = 15.781.



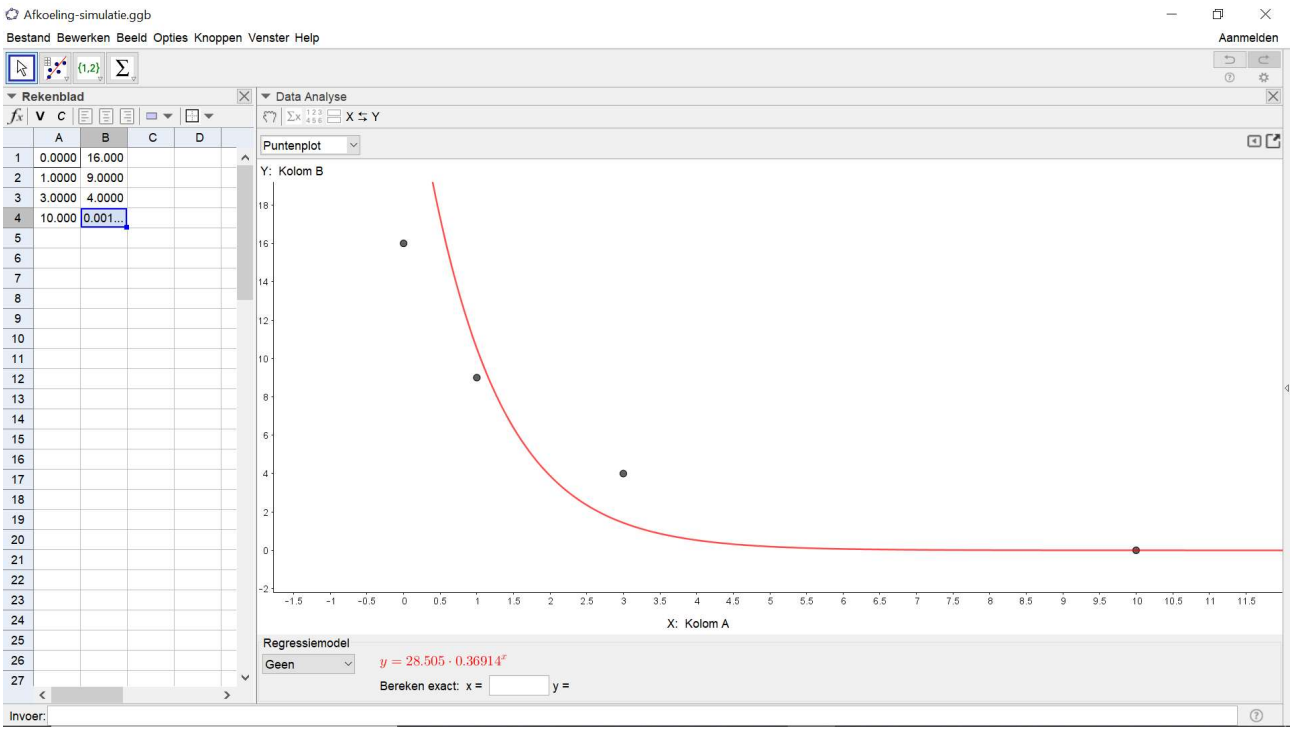
If you enter the same data in a TI-84 calculator, Graphmatica, GeoGebra, Wolfram, etc. they will just all refuse! Why? Theoretically, y can never be zero, so they assume it's safe to take log(y) to reduce the regression to a linear one. Now, practically, zero's *do* occur, if the y value is less than the smallest detectable value.

So what should we do in this case? This zero is very significant; it would be a waste to leave out 1 of 4 measurements! What I saw people doing: "Let's replace the zero by a 'very small number' and then it will work.". That seems to make sense, but actually, it doesn't! Should we replace the 0 by 0.1? 0.01? 0.001? Or should we just leave it out?

Let's see what happens:

		FittingKVdm	GeoGebra
0 replaced by 0.1	a	0.60529	0.60299
	b	15.778	16.181
0 replaced by 0.01	a	0.60491	0.47179
	b	15.781	21.477
0 replaced by 0.001	a	0.60488	0.37002
	b	15.781	26.309
last measurement omitted	a	0.60531	0.63508
	b	15.778	15.242

Wow! Apparently, the choice of that 'small number' doesn't have much influence on the parameter estimation using the iterative regression, but it has a dramatic impact in GeoGebra etc.: it can actually make the regression completely worthless, as you can see in this graph:



Like in the first example, the fourth point gets way too much weight because the logarithms were taken, and it pulls the whole curve in the wrong direction.

After all, it seems even better to just throw away the last measurement than to 'invent a small number'! But much better is: to use the iterative regression, clearly! Longer calculation times used to be a problem in the early days of computers, but nowadays, that's no longer an excuse for not using it!

P.S.: I work as a private math & physics tutor (and photographer), and I am available to assist you with data analysis and measuring methodology.
My software has another interesting feature: it can do 'multidirectional regression analysis'. It can be tried for free during 25 days. More information:
www.lerenisplezant.be/fitting.htm.
("Leren is plezant" is Flemish for "Learning is pleasant".)

Notes:

1 www.mathworld.wolfram.com/LeastSquaresFittingPowerLaw.html

2 www.itl.nist.gov/div898/handbook/pmd/section1/pmd143.htm

3 (log(1010) - log(990))/2 = 0.0043

References:

Dukkipati, Rao V.: "Numerical Methods", New Age International Publishers, 2010, ISBN: 978-81-224-2978-7