FittingKVdm – version 1.18.2

User manual – version 2024 04 26

Software developed by
Koen Van de moortel, experimental physicist (MSc)

Jules de Saint-Genoisstraat 98, 9050 Gentbrugge, Belgium
+32 9 2277036, mob.: +32 47 7368526
info@lerenisplezant.be
www.lerenisplezant.be/fitting.htm

Dear user, this program is not expensive, but it's not freeware. It took me a lot of time to program and to find or create all the examples!
I need your contribution to develop it (further)... Thanks for your understanding.
My bank account nr.: IBAN BE90 8907 1405 5932 (BIC VDSPBE91)

Contents

General description of this software program 6
The basic ideas. 6
Purpose: 6
Method: 6
Program requirements: 7
Your wishes... 7
Recommended way of working: 8
Entering data 11
1) Manually 11
2) From a file. 11
3) By generating artificial data. 13
Good advice. 14
Remarks about the report 15

Models available in this version 16
1) Polynomials 16
"Constant" 16
"Linear" 16
"Quadratic" 17
"Cubic" 17
"Orthogonal polynomial" (9th deg. Hermite) 18
2) Nonpolynomial monotonous functions - no asymptotes 20
"Power" 20
"Power with horizontal shift" 21
"Power-Möbius" 21
"Refractive index". 22
"Logarithm shifted through 0" 23
3) Nonpolynomial monotonous functions - 1 horizontal asymptote 24
"Exponential + baseline" 24
"Plateau - exponential" 25
"Diode" 26
"RLC serial filter - preset RL" 27
"RLC serial filter - preset RC" 27
"Weibull growth". 28
"Weibull decay". 29
"Rational 1". 30
"Rational 2" 30
"Parallax". 31
"Power" 31
4) Nonpolynomial monotonous functions - 2 horizontal asymptotes 32
"Logistic with baseline". 32
"Transition" 33
"Gompertz growth" 34
5) Peak functions. 35
"Gauss distribution" 35
"Added Gauss distributions". 36
"Dagum distribution" 38
"Double Logistic + baseline". 39
"Power•Exp.decay". 40
"Skewed peak 1 + baseline" 41
"Skewed peak 2 + baseline" 42
"Lorentzian peak + baseline" 43

"RLC serial filter" . . . 44
6) Other functions with 1 extremum . . . 45
"Chain line" . . . 45
7) Periodic and semiperiodic functions . . . 46
"Sine wave" . . . 46
"Added sine waves" . . . 47
"Damped sine wave" . . . 47
"Sine wave with harmonics" . . . 48
"Periodic peaks" . . . 49
"Skewed wave" . . . 50

About "goodness of fit" and "confidence intervals" . . . 51
Speed of convergence . . . 51
Chi-squared . . . 52
Use common sense! . . . 53
Residuals . . . 53
S versus parameter graph . . . 54
Confidence interval estimation . . . 55

Example data files (in alphabetical order) . . . 58
* adult men 11.6-13.8 percent fat - h vs m.dta1 . . . 58
* adult men 11.6-13.8 percent fat - m vs h.dta1 . . . 58
* Ages of married couples.dta1 . . . 59
* Air pressure vs Sun position 0-360 in Karlsruhe.dta1 . . . 60
* Air pressure vs Sun position in Karlsruhe.dta1 . . . 60
* Air pressure vs Sun position in Karlsruhe SD per degree.dta1 . . . 60
* Animal_metabolism_vs_mass_Kleiber_1932.dta1 . . . 62
* BMI2 vs fat Men 16 and older DIRECT DATA.dta1, BMI2 vs fat Men 16 and older.dta1 . . . 63
* BMI2 vs fat Women 16 and older DIRECT DATA.dta1, BMI2 vs fat Women 16 and older.dta1 . . . 63
* BMI3 vs fat Men 16 and older DIRECT DATA.dta1, BMI3 vs fat Men 16 and older.dta1 . . . 63
* BMI3 vs fat Women 16 and older DIRECT DATA.dta11, BMI3 vs fat Women 16 and older.dta1 . . . 63
* Boyle.dta1 . . . 64
* BUPA ALT vs GGT.dta1 . . . 65
* BUPA GGT vs ALT.dta1 . . . 65
* Car fuel consumption vs mass.dta1 . . . 66
* Ca-suspension.dta1 . . . 67
* Cell aptosis vs microwave field strength with rats (Karadayi 2024).dta1 . . . 70
* Cereal-crop-yield-vs-fertilizer-per-country.dta1 . . . 71
* Chain.dta1 . . . 72
* Charging_capacitor.dta1 . . . 73
* Child mortality UNICEF 1990.dta1 . . . 74
* Coin mass.dta1 . . . 75
* Concrete compressive strength vs age - no additives - 192 water.dta1 . . . 76
* Concrete compressive strength vs cement - no additives - age28.dta1 . . . 77
* Concrete-compressive-strength vs water-cement ratio age28 no add.dta1 . . . 78
* Cooling_of_water_cooker.dta1 . . . 79
* Cotton yield vs K conc Morteza Mozaffari.dta1 . . . 80
* Diode Si 1N4007.dta1 . . . 81
* Diode Ge OA72.dta1 . . . 81
* Diode White LED.dta1 . . . 81

* Driving times vs distance.dta1 . . . 82
* Electrical current from wind.dta1 . . . 83
* Eurovision Song Contest 2022 jury vs tele.dta1 . . . 84
* Eurovision Song Contest 2022 tele vs jury.dta1 . . . 84
* Examination scores vs completion time.dta1 . . . 85
* Extrav-Q4.dta1 . . . 86
* Extrav-Q49.dta1 . . . 86
* Extraversion Q4-Q49.dta1 . . . 86
* Falling pear.dta1 . . . 87
* Filter 1 RLC.dta1 . . . 88
* Filter 2 RL.dta1 . . . 90
* Filter 3 RC.dta1 . . . 91
* Food vs total expenditure 2017.dta1 . . . 92
* Friction of chairs on floor.dta1 . . . 93
* Happiness vs income.dta1 . . . 94
* Heart beat land mammals - Mortola 2015.dta1 . . . 97
* Heart rate after exercise.dta1 . . . 98
* Height_distribution_adults_USA.dta1 . . . 99
* Height_boys_12_18.dta1 . . . 100
* Height_boys_15.dta1 . . . 101
* Height_of_gutter.dta1 . . . 102
* Hot_stone_water.dta1 . . . 103
* Income distribution Belgium 2019.dta1 . . . 104
* Income distribution of households USA 2020.dta1 . . . 104
* Internal_resistance_9V_battery.dta1 . . . 106
* Janka hardness vs density of eucalypt wood.dta1 . . . 107
* JPG-Red_vs_coffee_concentration.dta1 . . . 108
* JPG-Green_vs_coffee_concentration.dta1 . . . 108
* JPG-Blue_vs_coffee_concentration.dta1 . . . 108
* Life exp vs health spending CH.dta1 . . . 109
* Life exp vs health spending JAPAN.dta1 . . . 109
* Life exp vs health spending USA.dta1 . . . 109
* Lifetimes_pressure_vessels.dta1 . . . 110
* Lynx.dta1 . . . 111
* Masks.dta1 . . . 113
* Mobile phone usage.dta1 . . . 114
* Noble prizes and chocolate.dta1 . . . 115
* Pendulum.dta1 . . . 116
* Pendulum variable length.dta1 . . . 117
* ping7.dta1 . . . 118
* Planet orbits.dta1 . . . 119
* Population_Nigeria1950-2022.dta1 . . . 120
* Powder flowrate (Flowlac90).dta1 . . . 121
* Pressure_vs_temperature_in_jar.dta1 . . . 123
* Pure water conductivity vs T.dta1 . . . 124
* Radon220 decay.dta1 . . . 125
* Refraction_polystyrene.dta1 . . . 126
* Rivers.dta1 . . . 127
* Running records.dta1 . . . 128
* Running speeds ultra-marathons men.dta1 . . . 129
* Running speeds ultra-marathons women.dta1 . . . 129
* Salt_water_G_vs_f.dta1 . . . 131
* Salt_water_conductivity_800Hz.dta1 . . . 131
* Shoe sizes adult men.dta1 . . . 132
* Shoe sizes adult women.dta1 . . . 132

* Smoking - adolescents Germany.dta1 134
* Sponge-ruler.dta1 136
* Sunspots-monthly avg.dta1 137
* Survival chances Men Belgium 2020.dta1 139
* Survival_friends-family.dta1 142
* Temperature Chatanga.dta1 144
* Temperature in coastal cities.dta1 145
* Temp Popovica 20230804.dta1. 147
* Throwing a pebble.dta1 148
* Tidal_current_Fundy.dta1 149
* Tumor growth Daskalakis 101.dta1 150
* Vocabulary vs age (Smith 1926).dta1. 152
* Vocabulary vs word count.dta1 153
* Vocabulary vs word count - De Maupassant-Short stories p1-80 & all.dta1. 154
* Wine ratings vs price Bordeaux 2018 RP.dta1. 157
* Wine rating comparison.dta1 159

General description of this software program

The basic ideas

Purpose:

To find a "best fitting" curve $y = f(x, p_1, p_2, p_3,...)$ (short: $y=f(x)$) of a specified kind through a set of n data points (x_i, y_i). This means: to adjust unknown parameters $p_1, p_2, p_3,...$ in the formula of f as good as possible.

Method:

"Regression": the wanted parameters are estimated by iteratively changing them so that the weighted sum of the squares of $f(x_i)-y_i$ (the vertical distances between the curve and each data point) are minimized. After each iteration, this sum is shown, divided by the degree of freedom, being the number of data points minus the number of parameters, in order to monitor the process. This is called the **"reduced χ^2 per degree of freedom"**. See: https://en.wikipedia.org/wiki/Reduced_chi-squared_statistic.

In the case of *invertible functions* (i.e.: f^{-1} is a function too), also the horizontal distances can be taken into account, by multiplying each term in this sum by $f^{-1}(y_i)-x_i$. This is new (2021) in this kind of software! I call this **"multidirectional regression"** (or fitting if you like).

So, what is minimized by adjusting the parameters is this:

* in the case of non-invertible functions: the traditional ("OLS" = Ordinary Least Squares) sum (with however the x errors added in the denominator if "Use σ_x for weights?" is checked):

$$S = \sum_{i=1}^{n} \frac{\left(y_i - f(x_i)\right)^2}{\sigma_{y,i}^2 \cdot \sigma_{x,i}^2}$$

* and in the case of invertible functions (if "Use multidirectional fitting" is checked):

$$S = \sum_{i=1}^{n} \frac{\left(y_i - f(x_i)\right)^2 \cdot \left(x_i - f^{-1}(y_i)\right)^2}{\sigma_{y,i}^2 \cdot \sigma_{x,i}^2}$$

(I abbreviate this as "MDLS" = Multidirectional Least Squares.)

More information:
www.researchgate.net/publication/350838636_Multidirectional_regression_analysis

The difference between OLS and MDLS can be visualised best with this simple example, a linear fit through the points (0, 0), (0, 10), (10, 0) and (10, 10):

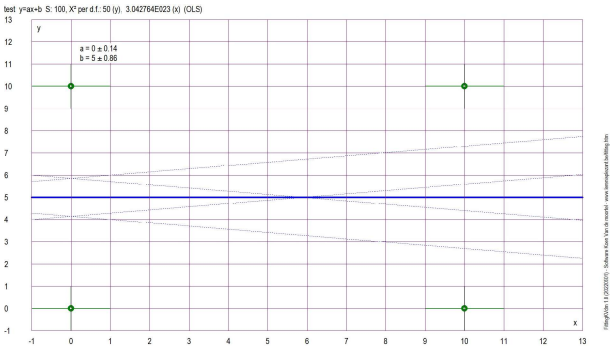


Figure 1 Linear OLS will fit a horizontal line through these four points. The sum of the vertical distances between the line and the points is minimized.

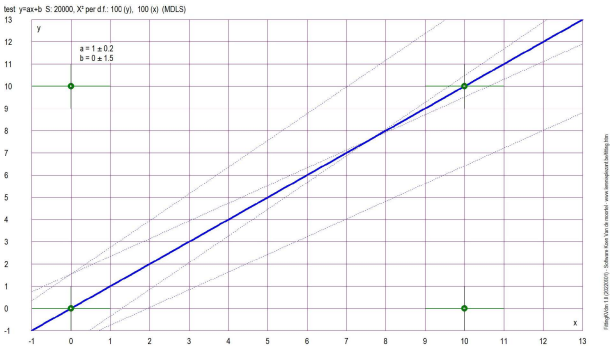


Figure 2 Linear MDLS will prefere this line. The sum of the products of the vertical and horizontal distances is minimal here.

The smart reader might notice that the MDLS sum would be the same if you drew the line from the upper left point to the lower right point. That's correct. That's what the algorithm will do if you start from different initial parameters. So, in this very weird case the MDLS sum has two minima, which might seem a "problem", but anyway it makes no sense at all to draw a conclusion from such data.

Program requirements:

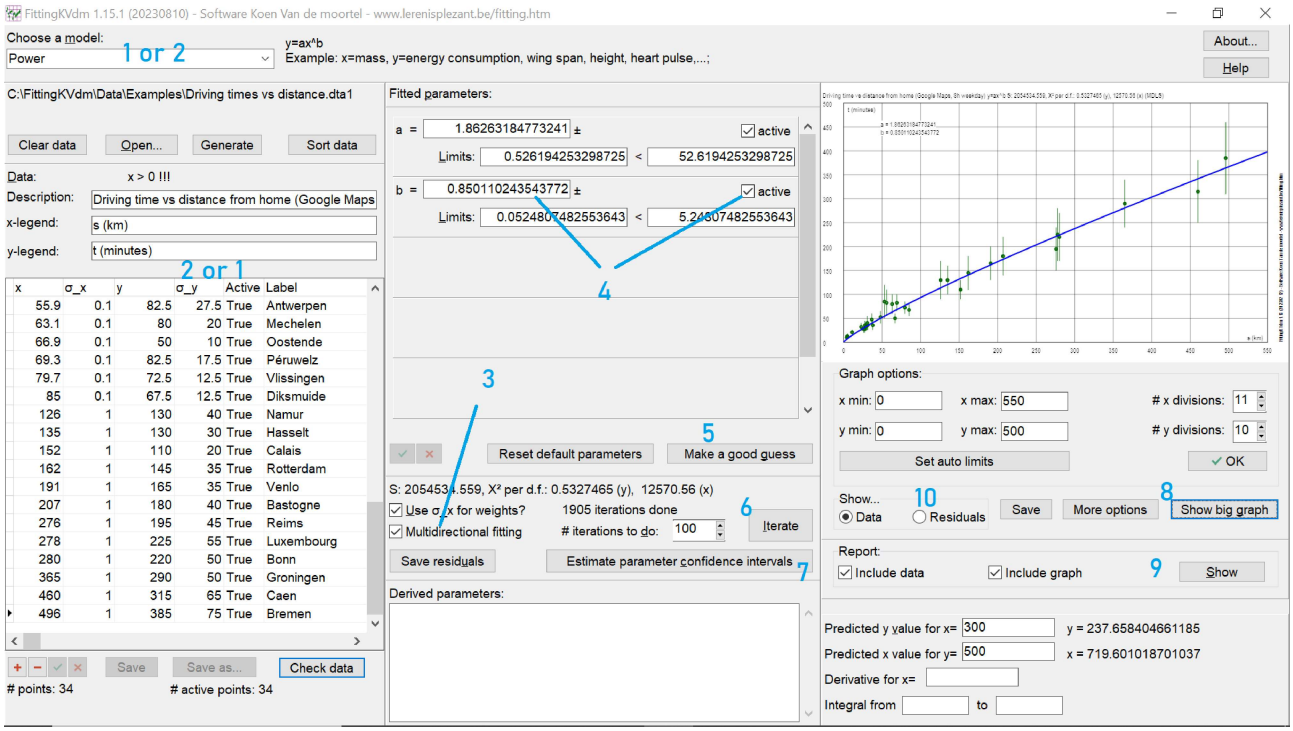
* Microsoft Windows operating system (any unicode version) or Mac with Windows emulator.
* The interface is optimized for the most common screen resolution nowadays (1920 x 1080 pixels), but it will work with less pixels too.

Your wishes…

Software is never "complete". If you think anything could be improved, like more models, more features, more documentation, other improvements,… please let me know! I plan to make regular updates. One of the things on my list is: multiple x variables.

My book **"Measuring and modeling by example - How mathematical functions can be used (and misused) to describe the world"** explains much more in depth about how to choose and judge models, and how to setup and analyze measurements.
See: https://www.lerenisplezant.be/fitting-book.htm

Recommended way of working:



1) **Enter data.** Of course, the number of data points should be at least the number of parameters in the model function. Details: see further.

2) **Select a model.** This should be compatible with your data, of course. E.g. you can not select a power function if there are negative x values in your data.
If you want to know how the model graph responds to parameter changes, just try some values. You can also adjust the values by clicking in the parameter edit box and then using the **"arrow up"** and **"arrow down"** keys ↑↓, which is much quicker!

3) **Select the method: OLS or MDLS.** If the model function can have extrema, you can only select OLS. Checking or unchecking "Multidirectional fitting" might give quite different results! (See further.)

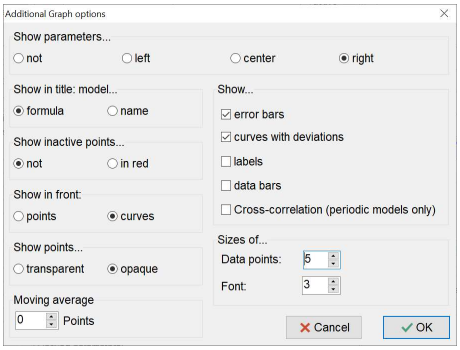
4) If you know that some parameter must have a specific value, like b = 0 if a linear model has to go through the origin, or a period T of a sine wave model, etc., you should enter the value and uncheck **"active"**. For most models, you can do this before the first guess.

5) Click the button **"Make a good guess"**. A possible problem with iteration is that if you start with bad initial parameter values or limits, there may be bad or no convergence. This software will help you out to make a reasonable estimation, based upon heuristic analyses specific for each model. This will usually work reasonably well if your data covers the typical features like asymptotes or peaks, and if the x values are in ascending order (if not: click the **"Sort data"** button).

6) Click **"Iterate"** several times (or enter a number of iterations to do) and check if the curve fits with the data points better and better. In some cases it may be necessary to change the lowest or highest allowed parameter value.
Remark: this button will only be enabled if **enough data points** were entered: at least one active point for each active parameter. If multiple points have the same x value, they are counted as one, since they can be replaced by one point: (x, weighted average of the y values). Example: normally two points are enough to define a line (y=ax+b), but two points like (5, 7) and (5,8) define a vertical line, which would mean a=∞, which is not allowed. If you see below the data grid: "# active points: 27 - 3" it means that there are 27 active points, but 3 have a duplicate x value.

7) After enough iterations, the parameters should stabilize, unless the dataset is crap or the model is not appropriate. Then you may click **"Estimate parameter confidence intervals"** to find out how reliable the parameters are. (The button is enabled after 10 iterations, but it's usually better to do some more iterations first.)

8) Set the graph options according to your wishes and click **"Show big graph"** to admire the beauty of the fitted curve. This graph is automatically temporary saved as "temp.svg" in the program map, but you can save it with a more appropriate name if you click the "**Save**" button below the graph. Since version 1.13 this is a "**Scalable Vector Graph**" (SVG), witch can be opened in recent browsers, drawing programs like InkScape, or word processors. You can also save it as a JPG using the popup menu when you right-click the big graph, but the SVG has the advantage that it can be resized without quality loss.
You can zoom in on details in the graph, using two fingers on your mousepad.
Hover over a data point to see its coordinates and label.



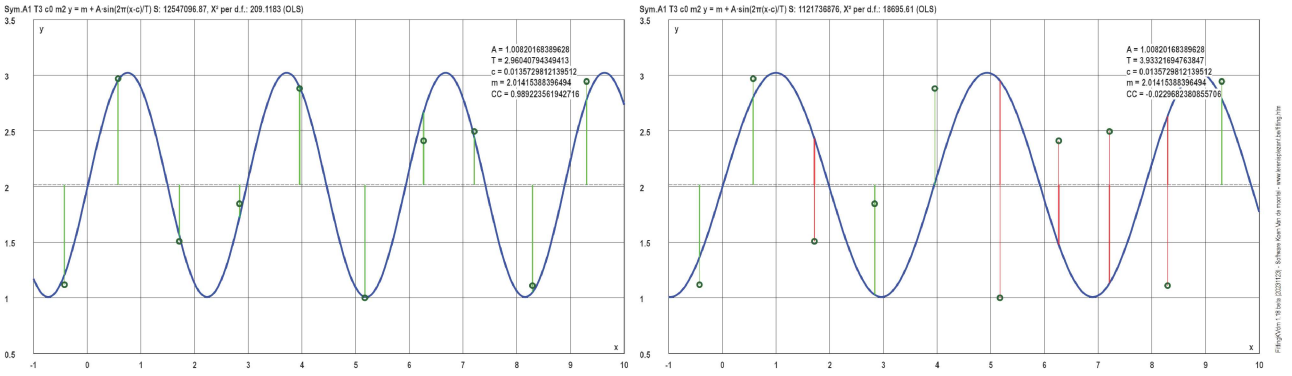
Click the "**More options**" button if you want to want to fine-tune the graph (adjust the parameter position, the font size, etc.). One of the options is to set the data points transparent; in that case, multiple identical data points appear darker (see the "shoe size" example.

Another interesting option is the number of **"Moving average"** points. If you set this to, let's say 5, a faint blue line will be shown from which the first point is the weighted average of the first 5 points; the second point is the average of points 2..6, etc.
If you set the value to 0, nothing is shown. A value of 1 produces a simple connection between the measurement points. When using MDLS, meaning x and y could be switched, also a green line will show up: these are the moving averages calculated vertically. If there is a good fit, the blue and the green lines should mostly coincide.

If you want to see a **histogram** rather than a scatterplot, check "data bars" and set "size of data points" to 0. See the example file "Height_boys_15.dta1".

For the study of data with a **periodicity**, you might want to check the option to vizualize the "**cross-correlation with the sine function**" (c). The green lines show up when the measurement and a sine wave with the given period and phase are on the same side as the given base line. If both are on different sides, the line is red. All

green lines mean there is a good match of both period and phase (c≈1). All red lines mean that the period matches but the phase is 180° off (c≈-1). If there is is no good match, c≈0.
See: https://en.wikipedia.org/wiki/Cross-correlation
The graphs below show a good and a bad cross-correlation.

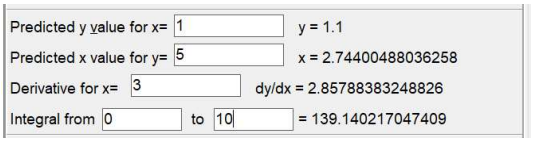


9) If you want a HTML report, you may now click the "Show" button in the **Report** section down right. The last shown graph will appear there. The report with an embedded SVG image of the graph will be saved automatically in the same folder as the data.

10) You may want to do some more investigation of the model, for example by looking at the **residuals graph** (differences between measured and calculated y values). (See further.)

You can switch (1) and (2). If no data have been entered and you select a model, you will see some typical example parameters and the corresponding graph. Play with the parameters to get a feeling of how they influence the shape of the curve.

Remark: if you started with OLS, hit the "Guess" button, and then select MDLS and iterate, it's possible that the program switches back to OLS, because the parameter limits were to wide. If you select MDLS and then click "Guess", the limits will be set safely, is possible.



Additionally, you can enter an x value to **predict y**, and with invertible functions also y to **predict x**. Since version 1.9, you can have the **derivative** calculated for a given x, and the **integral** between given boundaries. The integral is calculated analytically (using the primitive function) for these models: "Constant", "Linear", "Quadratic", "Cubic", "Power", "Exponential + baseline", "Plateau - exponential", "Rational 1", "Rational 2", "Sine wave", "Added sine waves", "Damped sine wave", "Sine wave with harmonics", and numerically in other cases. This allows you, e.g. to get the work after measuring force and distance, or the total electrical charge after measuring current vs time, etc.

Entering data

1) Manually

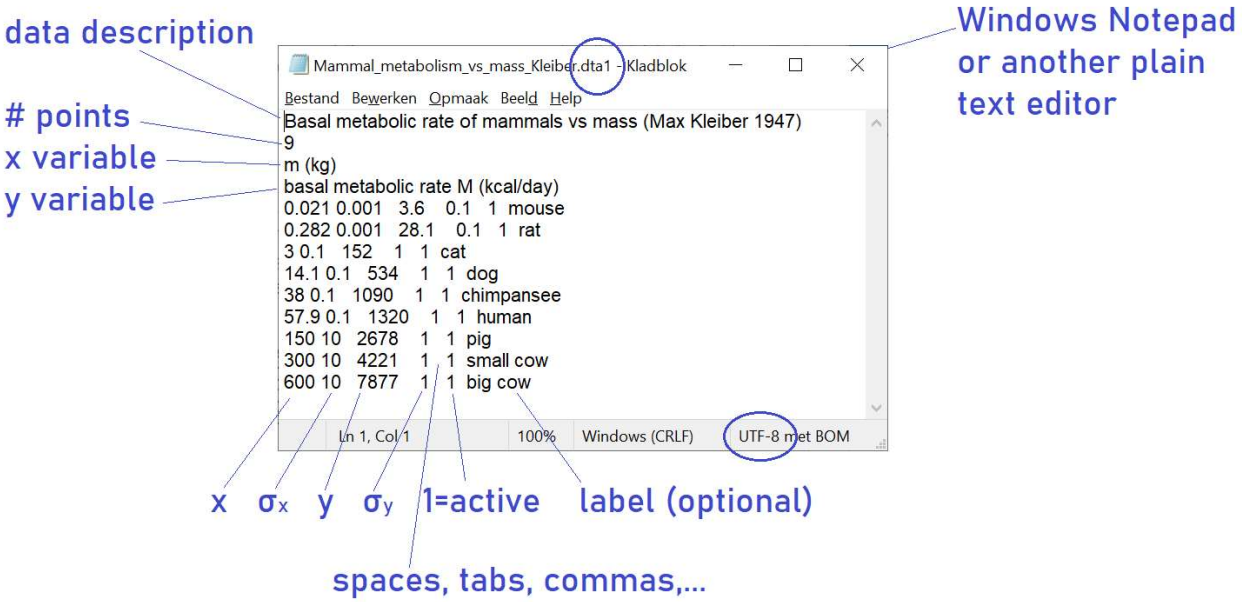
You can enter numbers manually in the data grid. Essential are of course: the x and y values of all your measurements, but also the estimated precisions σ_x and σ_y, ("errors" or "confidence intervals" if you will) of the values are required. Zeros are not allowed! There is no such thing as a zero error on measurements. If you don't know them, you might enter all "1" values (= default), but we recommend to use realistic values. E.g. if you measure a voltage value "3.56", with a resolution of 0.01, the precision will be in the same order, but usually it's a bit worse, like 0.02 or so. See the manual of your instrument.
If your y values are **counts in a sample of a population** (for example the number of people between 160 and 165cm tall), usually $\sigma_y=\sqrt{y}$ can be entered. If the number in a certain class (bin) is zero, that rule would suggest $\sigma_y=0$, but that is unrealistic. There seems to be no agreement about what to do in this case; if the distribution is not yet known, it's impossible to know σ_y. I suggest using 1, which might be too big, but better too big than too small. This will cause the better known values to have mor weight.
If y is a **fraction** (for example the percentage of vegetarians in a population divided by 100), a good estimate for $\sigma_y=\sqrt{(y\cdot(1-y)/n)}$, with n the sample size.
The values of σ_x and σ_y will be visible in the error bars in the graph, but more importantly, they will be used to weigh the importance of each data point and to calculate the χ^2 values that give you an idea of the goodness of the fit.
When you enter data, the default value for "active" will be "true" (the box in that column is checked). That means that this point will be used in the calculations. If for some reason, you decide that the point is bad (mistake, erroneous measurement, outlier), you can uncheck the box.
Optionally, you can enter a descriptive label for each point. This may be shown on the graph if you select that option.

Above the data grid, you can enter a description of the data set (title in the graph), a description of the x variable, which will be shown on the x axis of the graph, and a description of the y variable (y axis of the graph).

2) From a file

You may want to copy and paste from a spreadsheet or another program, or from a CSV file. For that, you have to create a simple plain text file with Windows Notepad (or another editor). Make sure that, before you copy, next to the x and y columns, you also have columns with σ_x and σ_y, and column with "1" or "0" values (active or omitted). A column with labels is optional.
Before the data columns, you need to insert 4 lines with information (see below).
Save the file as unicode (UTF8) text (normally the default setting) with extension ".dta1".
(In a later version, ".dta2" will be for 2 x variables etc.)

The structure of the data files is as you see in the following example:



Line 1: description of the data set;
Line 2: number of data points (If you would change the 9 to 6, the last 3 measurements would be ignored.);
Line 3: description of the x variable;
Line 4: description of the y variable;
Next lines (max. 50000): x, σ_x, y, σ_y, "1" or "0" if this data point is "active" or "inactive", and optionally a label (max. 100 characters).
Important: use points, not commas as decimal separator! You can use commas to separate different numbers; don't use them in numbers. "123,456" will be read like two numbers "123" and "456"! If one your variables is a date, enter the values as the number of days since a certain starting date, not like "12/4/2021" or something like that. In spreadsheet programs like Excel or Quattro Pro there is a function to do that. You might enter the dates in the "label" column if you wish, but this format can not be used for calculating.

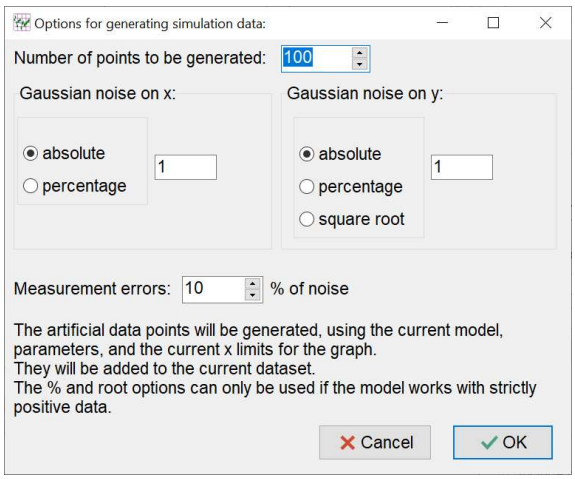
https://rforbiochemists.blogspot.com/2015/06/are-my-fitted-lines-significantly.html					
21	# measurements				information
substrate conc (mM)					
v (µM/min/mg)					
2	0.01	59.01	0.01	1	WT
1	0.01	58.29	0.01	1	WT
0.6	0.01	54.17	0.01	1	WT
0.5	0.01	51.82	0.01	1	WT
0.4	0.01	49.76	0.01	1	WT
0.3	0.01	45.45	0.01	1	WT
0.2	0.01	36.88	0.01	1	WT
0.1	0.01	26.1	0.01	1	WT
0.09	0.01	23.5	0.01	1	WT
0.08	0.01	22.26	0.01	1	WT
0.06	0.01	16.45	0.01	1	WT
0.04	0.01	13.67	0.01	1	WT
0.02	0.01	6.14	0.01	1	WT
0.05	0.01	11.8	0.01	1	H297F
0.1	0.01	19.9	0.01	1	H297F
0.2	0.01	30.3	0.01	1	H297F
0.3	0.01	36.6	0.01	1	H297F
0.4	0.01	40.2	0.01	1	H297F
0.5	0.01	42.1	0.01	1	H297F
1	0.01	47.8	0.01	1	H297F
2	0.01	50	0.01	1	H297F
x	precision	y	precision	1=active	label

x and y legends

Example of data preparation in a spreadsheet program.

Select the block, copy and paste it in Notepad and save it as a text file with extension ".dta1".

3) By generating artificial data



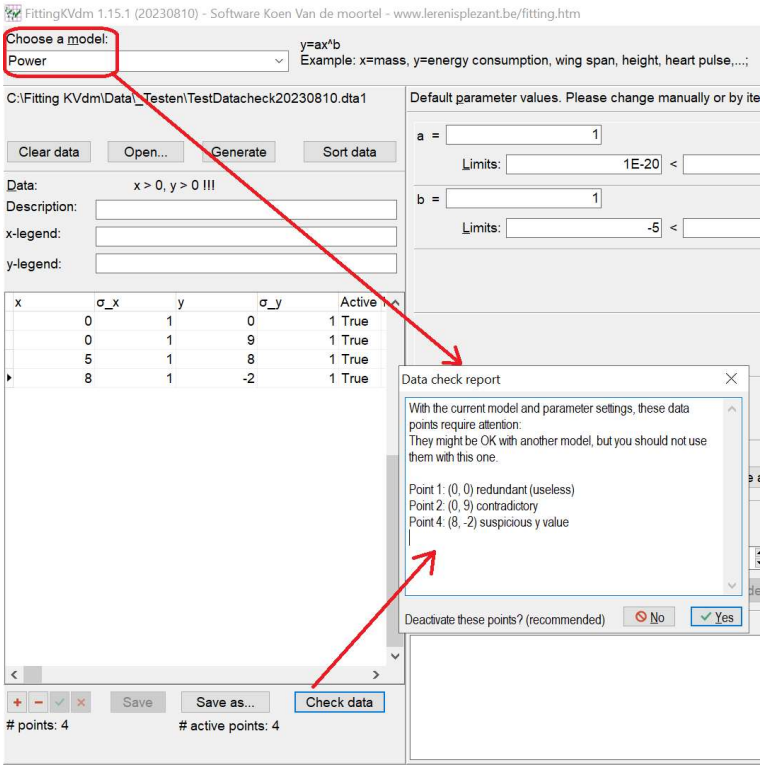
Click the "Generate" button, and enter how many data points you want to add. These will be invented by choosing this number of x values in the current graph window (uniformly distributed), and by calculating the corresponding y values, and adding gaussian noise to the x and y values, i.e. random numbers from a normal distribution with the chosen standard deviation. The amplitude of the noise has to be given in absolute value or as a percentage of the x or y value (only possible for models that allow no negative values). If y is a counted number, you should use the option "square root". The measurement errors (size of the error flags) can be chosen as a percentage of the noise amplitude.

Good advice

* Always try to **use raw data**, don't "smooth" or filter, avoid any form of "preprocessing", especially taking logarithms! See: https://www.researchgate.net/publication/349324179_Non-linear_regression_-Why_you_shouldn't_take_the_logarithms_of_your_variables
  Only in case you fit distributions (Gauss,...) and x is a continuous variable, you should make classes first if you want to obtain the classical Gauss curve.
  And for exponential and periodic models it makes sense to subtract a large part of the x variable, e.g. enter the time in years since 1990: 0, 1, 2, 3... instead of 1990, 1991, 1992, 1993,... because small errors in the growth factor or period will cause huge deviations if you apply big exponents or calculate many periods further.

* Don't leave points out just because they don't "look good" (**"outliers"**), only if you have good reasons to believe something went wrong with the measurement, or if a physical/logical reason indicates it doesn't belong in the dataset.

Some data points make no sense, e.g. a negative y value for a model that expects positive values.
Also, it's possible to have totally useless points. Suppose you enter (0,0) and try to fit a "power" model with a positive exponent. In this case f(0)=0 always, no matter what the parameters values are, so this point should be left out (set inactive) because it doesn't help the fitting. A data point (0,5) should also be left out here, since it contradicts with the function.
The same can be said about the "linear" model with b set inactive (fixed to a certain value), or a "sine wave" model with the phase shift set fixed, etc.



Click the "Check data" button to automatically search for such points.

Remarks about the report

The **averages** and **standard deviations** of both x and y are calculated without and with weights ($1/s^2$ with s the uncertainty for each x or y), using the traditional formulas.

The **quantiles** (q_i) are calculated with the most logical formula:

$$q_i = x_k + \alpha \cdot (x_{k+1} - x_k)$$

with

$$k = \left\lfloor \frac{n}{4} \cdot i + \frac{1}{2} \right\rfloor = \text{floor}\left(\frac{n}{4} \cdot i + \frac{1}{2} \right)$$

$$\alpha = \frac{n}{4} \cdot i + \frac{1}{2} - \left\lfloor \frac{n}{4} \cdot i + \frac{1}{2} \right\rfloor = \text{frac}\left(\frac{n}{4} \cdot i + \frac{1}{2} \right)$$

and x_1 .. x_n the data points, in ascending order.

The same is done for the y values.
(q_1 = first quartile, q_2 = **median**, q_3 = third quartile)

For example, if n = 20:

$$q_1 = x_5 + \frac{1}{2}(x_6 - x_5) = \frac{x_5 + x_6}{2}$$

If n = 21:

$$q_1 = x_5 + \frac{3}{4}(x_6 - x_5)$$

The version of **Kendall's tau** that is calculated, is the most simple and logical one:

$$\tau = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{sgn}((x_i - x_j)(y_i - y_j))$$

with n the number of active data points; "sgn" is the signum function (-1, 0 or 1 if the argument is negative, zero or positive).

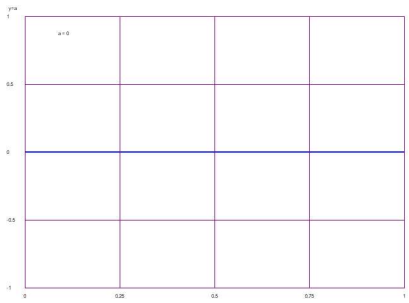
Interpretation: if the row of data points is perfectly ascending, $\tau = 1$, if it is perfectly descending, $\tau = -1$. If the data points are chaotic or going up and down, τ is much more near zero.

Pearson R (= r_{xy}) is calculated as described in every textbook, or e.g.
https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Models available in this version

1) Polynomials

"Constant"



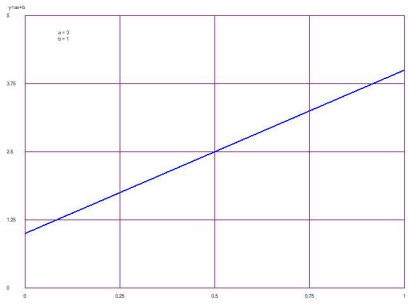
$$f(x) = a$$

Features:
Horizontal line

"Make a good guess" will show the average y value; iteration will bring you to the weighted average.

Multidirectional fitting?
No, it makes no sense, since an infinite number of x values are possible for the one y value!

"Linear"



$$f(x) = ax + b$$

Constraints:
a and b can be anything, but if a=0, f is not invertible, and you better use the "constant" model.

Features:
Straight line, ascending if a > 0, descending if a < 0.
Intercept with vertical axis: b. So, if you are sure that a zero x value has to give a zero y value (e.g.: x = number of apples, y = price), set b fixed to zero.

Derived parameter:

* $z = -\frac{b}{a}$

(the zero value)

Guess:
Line through the first and the last point.
All parameters can be fixed (deactivated) before the guess.

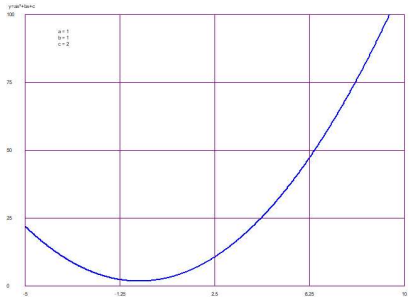
Examples:

* If x = temperature and y=pressure in °C or °F of a gas in a closed jar, z is the experimental value absolute of the absolute zero temperature (0K) in °C or °F.
  See the file "Pressure_vs_temperature_in_jar.dta1".

* Many physical relationships are linear in approximation, like e.g. if y = F (force) and x the extension or compression of a spring, the parameter "a" would be the spring constant and b = 0 (Hooke's law).

Multidirectional fitting?
If x and y are really causally connected like in the examples above, use it!
If you don't have a clue about a causal connection and you just want to see if there is some kind of trend going on, don't use it. For example: x=time and y=average annual quantity of beer per person consumed in a certain country.

"Quadratic"



$$f(x) = ax^2 + bx + c$$

Constraints:
If a = 0, better use the "linear" model.

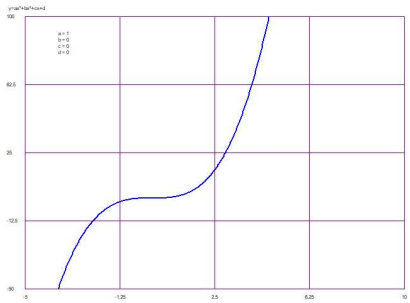
Features:
Parabola with "valley" shape if a > 0, "hill" shape if a < 0;
Min./max. if x=-b/(2a).
Zeroes if $D=b^2-4ac>0$: $x = (-b\pm\sqrt{D})/(2a)$
If the curve has to go through the origin, set c = 0.

Guess:
Parabola through 1st, middle & last point.

Example:
* If x = time and y the distance an object has fallen without friction, a = g/2 (half of the gravitational acceleration), b = v_0 (starting speed) and c = y_0 (starting height). See: https://en.wikipedia.org/wiki/Free_fall; example data file: "Falling pear.dta1".

Multidirectional fitting?
No; 0 or 1 or 2 x values are possible for each y value!

"Cubic"



$$f(x) = ax^3 + bx^2 + cx + d$$

Constraints:
None, but if a=0, use "quadratic".

Features:
Curve goes from $-\infty$ to $+\infty$ if a > 0, otherways if a < 0.
1, 2 or 3 zeroes

0 or 2 extrema
1 inflection point in x=-b/(2a).
If the curve has to go through the origin, set d = 0.

Guess:
Curve through 4 evenly spread points.

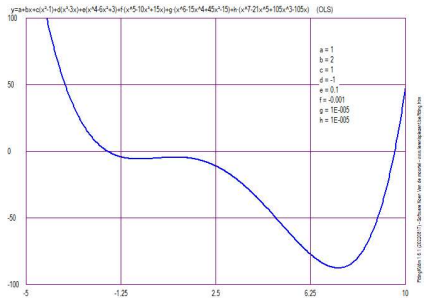
Examples:
* You can fit a 3rd degree polynomial through any set of data, but it's usually nothing more than a "passe-partout" model in case you have no clue what the real connection between x and y is.

Multidirectional fitting?
No; multiple x values may be possible for each y value!

"Orthogonal polynomial" (9th deg. Hermite)

$$f(x) = a + b\cdot x + c\cdot\left(x^2-1\right) + d\cdot\left(x^3-3x\right) + e\cdot\left(x^4-6x^2+3\right) + f\cdot\left(x^5-10x^3+15x\right)$$
$$+ g\cdot\left(x^6-15x^4+45x^2-15\right) + h\cdot\left(x^7-21x^5+105x^3-105x\right)$$
$$+ i\cdot\left(x^8-28x^6+210x^4-420x^2+105\right) + j\cdot\left(x^9-36x^7+378x^5-1260x^3+945x\right)$$

Constraints:
None. To avoid confusion: a, b...j are real numbers here.



Features:
N-th degree polynomial can have n-1 extrema, maximally n zeroes.
If n is odd: curve goes from -∞ to +∞ if the coefficient of the highest degree > 0, otherways if it is < 0.
If n is even: both limits are +∞ if the highest degree has a positive coefficient, otherways they are both -∞.

Guess:
All parameters are set to zero, only the upper and lower boundaries are estimated. These are good starting values in this case.

Examples:
No natural law can be written as a polynomial with such a high degree.
So, the usage of this model function is only to do approximations within the domain of the collected data, if you have absolutely no clue about the real functional relationship between x and y, and you want to see *some* curve. The more the data points are nicely spread out, the better this will work. But, you should never thrust extrapolations of polynomial models, never.

Multidirectional fitting?

No; multiple x values may be possible for each y value!

Remarks:

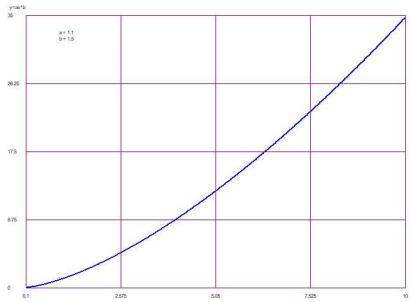
* Why not just use a simple polynomial like $y=a+bx+cx^2+dx^3+...$?
  The reason is simple: the convergence will be much more stable. Usually the absolute values of the parameters will be a nicely descending row. Using a simple polynomial, if you fit the first parameters, and then you add the next term, all the previous parameters might completely flip. This will be much less when you use orthogonal polynomials, like the Hermite set.
* If you don't have many data points, you can deactivate the highest powers of x, to avoid "overfitting". If you expect your data to have a symmetrical pattern left and right of the y axis, you should only make a, c, e, g and I active; if the origin is supposed to be a symmetry center, only activate b, d, f, h and j.

More background information:
https://en.wikipedia.org/wiki/Hermite_polynomials

2) Nonpolynomial monotonous functions - no asymptotes

"Power"

$$f(x) = ax^b$$



Constraints:
a > 0, b can theoretically be anything, but if b=0 there is not much fun and the function is not invertible; $x_i \geq 0$, $y_i \geq 0$.

Features:
Curve goes through the origin if b > 0.
Usually this model function is used with b > 0, and then it has no asymptotes, but you can use it with b < 0, and then it has a vertical asymptote y = 0 and a horizontal one x = 0 (see further).
Special case: b = 1, linear through the origin.

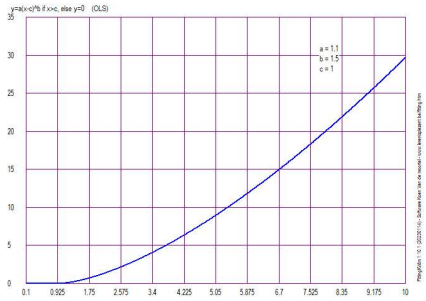
Guess:
Calculated from 2 points.
All parameters can be fixed (deactivated) before the guess.

Examples (with b>0):

* Scaling laws for objects/bodies with similar density distributions (=hypothesis): x = height or some other size, y = mass, b should be 3 if the hypothesis is true.
  See the example files "adult men 11.6-13.8 percent fat - h vs m.dta1" and "adult men 11.6-13.8 percent fat - m vs h.dta1", "Animal_metabolism_vs_mass_Kleiber_1932.dta1".
* Time (y) needed to run a distance x. If the runner wouldn't get tired, b would be 1, but in reality b>1; see: "Running records.dta1".
* Time needed for a distance on the road; see "Driving times vs distance.dta1"
* Orbit periods of the planets vs their distances (semi-major axis) to the Sun (Kepler's third law), see: "Planet orbits.dta1".
* Very roughly: flow rate vs length or drainage basin surface of a river; see: "Rivers.dta1"
* Approximately: x = # words in a text, y = # *different* words in the text. See the file "Vocabulary vs word count.dta1".
* The force required to squeeze a sponge versus the distance it was pressed; see: "sponge-ruler.dta1".

Multidirectional fitting?
Definitely use it! The results will be dramatically better than without using it!

"Power with horizontal shift"



$f(x) = a \cdot (x - c)^b$ if x>c, otherwise y=0

This is kind of an artificial expansion of the previous model, shifted to the right by c if c > 0.

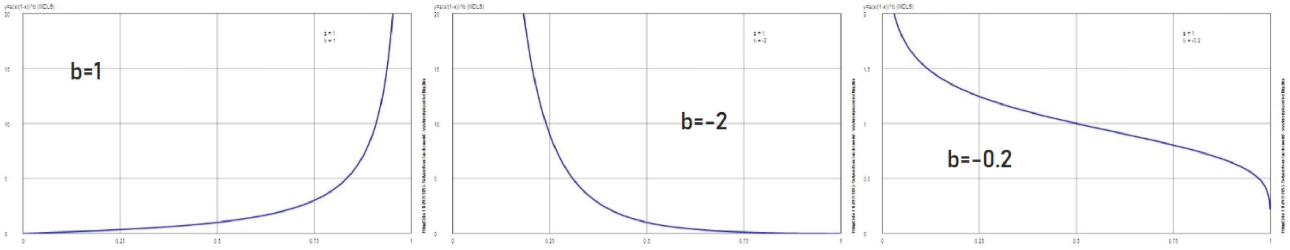
Examples:

* Simplistic model for flow rate of a powder with particle size c, through a hole with diameter x. See file: "Powder flowrate (Flowlac90).dta1".

Multidirectional fitting?
Yes, as long as the x measurements are bigger than c.

"Power-Möbius"

$$f(x) = a \cdot \left(\frac{x}{1 - x} \right)^b$$



Constraints:
$a \geq 0$; $0 \leq x_i < 1$, $y_i \geq 0$

Features:
If b>0: curve through the origin, ascending and going to a vertical asymptote at x = 1.
If b<0 the curve is mirrored horizontally.
If |b|<1 it starts/ends vertically.

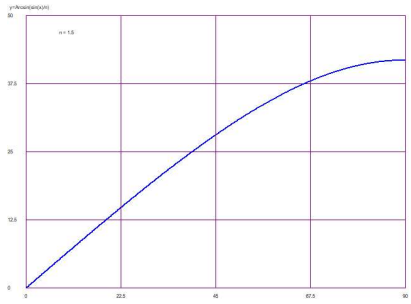
Guess:
Calculated from two points.

Possible applications:
Suppose you add X units of cement to 1 unit of sand and the strength of the concrete $S \sim X^b$, then the percentage of cement is x=X/(1+X), so X=x/(1-x), and $S=a(x/1-x))^b$.

Multidirectional fitting?
Yes, use it!

“Refractive index”

$$f(x) = \mathrm{Arcsin}\left(\frac{\sin(x)}{n}\right)$$



Constraints:
$n \geq 1$, $0 \leq x_i < 90°$, $y_i \geq 0$

Features:
Curve through the origin, ascending and ending flat at x = 90°.

Guess:
Calculated from one point.

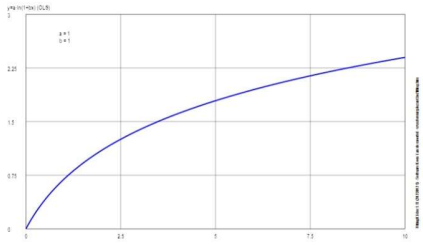
Example:

* x = angle of incoming light beam (°), y = angle of refracted beam, n = the desired relative refractive index of the test substance.
  See example file: “Refraction_polystyrene.dta1”

Multidirectional fitting?
Yes, use it!

“Logarithm shifted through 0”

$$f(x) = \frac{a}{b} \cdot \ln(1 + bx)$$



Constraints:
$x_i \geq 0$, $y_i \geq 0$, $a \geq 0$, $b > 0$

Features:
This is a linear transformation of the natural logarithm function. The extra b under a wasn’t really necessary, but it makes the interpretation easier. The curve goes through the origin with maximal slope (f’(0) = a), and it keeps ascending slower and slower, but without horizontal asymptote. The vertical asymptote at x=-1/b can be ignored if you use this function with positive variables.

Guess:
Take the leftmost point (x_1, y_1) and the rightmost (x_2, y_2). Then $a \approx y_1/x_1$. This can be a rough approximation, but it’s close enough to get the iteration starting.
Parameter b can be estimated from

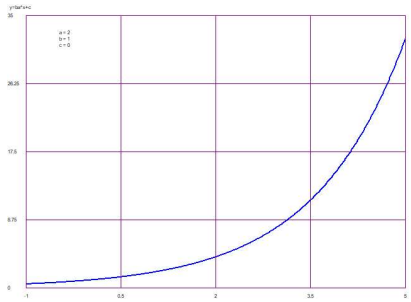
$$\frac{y_2 - y_1}{x_2 - x_1} \approx f'\left(\frac{x_1 + x_2}{2}\right) = \frac{a}{1 + b\frac{x_1 + x_2}{2}}$$

Example files:
* “Food vs total expenditure 2017.dta1”
* “Vocabulary vs word count - De Maupassant-Short stories p1-80 & all.dta1”.

Multidirectional fitting?
Yes, use it!

3) Nonpolynomial monotonous functions – 1 horizontal asymptote

"Exponential + baseline"



$$f(x) = b \cdot a^x + c$$

Constraints:
a>0

Derived parameter:
* Half-life or doubling time:

$$T = \left| \frac{\log(2)}{\log(a)} \right|$$

Features:
Horizontal asymptote y = c.
Always ascending if a > 1, always descending if a < 1.

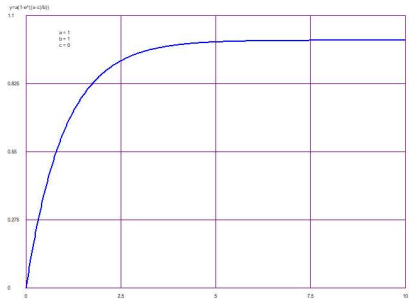
Guess:
Calculated from the first, middle and last point.
If the data seem to make no sense: a = 1, b = 0, c = $y_{average}$.
All parameters that can be fixed (deactivated) before the guess.

Examples:
* Newton's cooling law: x = time, y = temperature of an object; c = environment temperature, b = temperature excess at t = 0, 0<a<1, small a = fast cooling. See example file "Cooling_of_water_cooker.dta1".
* Growth of a population that has unlimited space and food; x = time, y = number (population size), b = start size, a>1, c=0. See file "Population_Nigeria1950-2022.dta1".
* Radioactive decay of a certain isotope: x = time, y = mass of the isotope that is left after x time, b = start mass, c=0.
* Charging/discharging of a capacitor (b<0/b>0): x = time, y = voltage.
* Dimming of light due to absorption (x = distance, y = intensity).
* Heart rate recovery, see example file "Heart rate after exercise.dta1".
* A learning effect; see "Throwing a pebble.dta1".

Multidirectional fitting?
Same remark as with the "Linear" model, and also the expected c should be out of the range of all y values. E.g. if the curve is descending towards the baseline (a<1 and b>0), the upper limit for c should be less than the lowest y value. If not, use OLS.

“Plateau - exponential”

$$f(x) = a \cdot \left(1 - e^{-\frac{x-c}{b}}\right)$$



Constraints:
$a>0$, $b>0$; $y_i \geq 0$

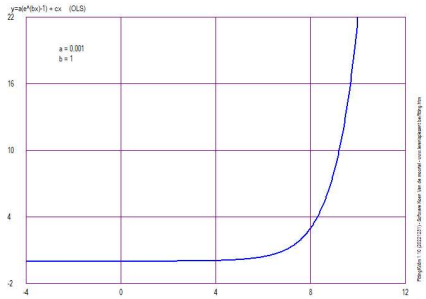
Features:
This model overlaps partially with the other exponential model, but the parameters are expressed in a different way, which might sometimes be more convenient.
Horizontal asymptote: y = a.
If you want the curve to go through the origin, set c = 0.

Guess: a = slightly more than y_{max}; c = 0; b: calculated from a, c and the middle point.
All parameters can be fixed (deactivated) before the guess.

Example:
* x = time, y = voltage over a capacitor that is charged with a source with voltage a over a resistor with resistance R, b = RC, c = starting time of the measurements.
  See file “Charging_capacitor.dta1”.

Multidirectional fitting?
This function is invertible as long as the y values are within]0, a[. If you set the lower limit of the a parameter a little bit above the highest y value, you can use multidirectional fitting to improve the results.

“Diode”



$$f(x) = a \cdot \left(e^{bx} - 1\right)$$

Constraints:
a > 0, b > 0

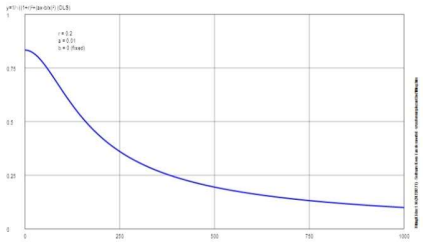
Features:
This is an ascending exponential function, but shifted down so it always goes through the origin.
Horizontal asymptote: y = -a.

Guess: a and b are calculated from the middle and the rightmost point (neglecting the “-1” in the equation, which is okay in real situations).
All parameters can be fixed (deactivated) before the guess.
This model is very difficult to fit if the start values are not well chosen!

Example:
* x = voltage over an “ideal” diode, y = current through it, at constant temperature.
  The maximal reverse current is -a.
  Parameter b = q/nkT (q = elementary charge, k = Boltzmann constant, T = absolute temperature, n = ideality factor ≈ 1..2).
  See: https://en.wikipedia.org/wiki/Shockley_diode_equation
  and the example file “Diode Si 1N4007.dta1”.

Multidirectional fitting?
This function is invertible as long as the $y_i > -a$. If you set the lower limit of the a parameter a little bit below the lowest y value, you can use multidirectional fitting to improve the convergence.

"RLC serial filter - preset RL"



This is the same as "RLC serial filter", but with a fixed parameter b = 0.

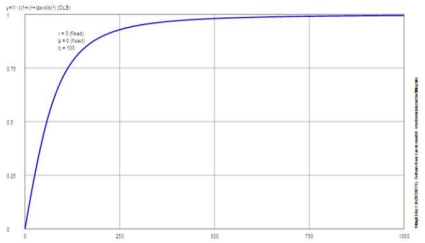
Features:
Descending curve; horizontal asymptote y = 0.

Examples:

* x = frequency, y = U_{out}/U_{in} with U_{in} the AC voltage over a coil (inductance L in henry, internal resistance R_i in ohm) + resistor (resistance R in ohm) in series; U_{out} is measured over the resistor ("low pass RL filter). The fitted parameters b and r are related to L and R_i; see the "RLC serial filter" model. See example file "Filter 2 RL.dta1".

Multidirectional fitting?
Yes, as long as the y values are within]0, 1[.

"RLC serial filter - preset RC"



This is the same as "RLC serial filter", but with fixed parameters r = a = 0.

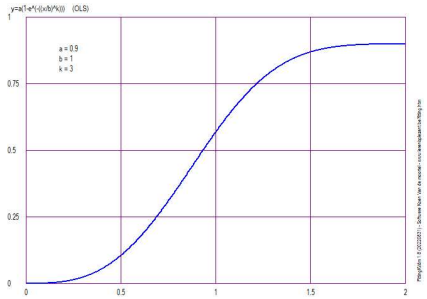
Features:
Curve ascending from origin; horizontal asymptote y = 1.

Examples:

* x = frequency, y = U_{out}/U_{in} with U_{in} the AC voltage over a capacitor (capacitance C in farad) + resistor (resistance R in ohm) in series; U_{out} is measured over the resistor ("high pass RC filter"). The fitted parameter b is related to C; see the "RLC serial filter" model. See example file "Filter 3 RC.dta1".

Multidirectional fitting?
Yes, as long as the y values are within]0, 1[.

"Weibull growth"



$$f(x) = a \cdot \left(1 - e^{-\left(\left(\frac{x}{b}\right)^k\right)}\right)$$

Constraints:
$a > 0$ ("amplitude" of the transition);
$b > 0$ (determines the steepness);
$k > 0$.
$x_i > 0$, $y_i \geq 0$

Features:
Curve through origin; horizontal asymptote y = a.
Inflection point exists when $k > 1$.

Guess:
$a \approx y_{max}$
The x position where y is at half the max. height (x_{half}) is estimated; then

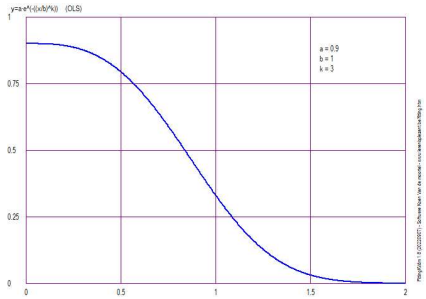
$$b = \frac{x_{half}}{\sqrt[k]{\ln 2}}$$

k = 2 to start. It is advized to keep it fixed until a and b are more or less "settled".
All parameters can be fixed (deactivated) before the guess.

Examples:
* x = time, y = concentration of suspended material; see: www.researchgate.net/publication/361983006_Nonlinear_Models_by_Parameters_and_their_Transformation; example file: "Ca-suspension.dta1".
* Interpreted as a cumulative distribution, it can describe the % or number of deceased persons or broken devices vs age. If y is a percentage, set a = 1, otherwise a = total number of persons/devices. Example file: "Lifetimes_pressure_vessels.dta1". More: en.wikipedia.org/wiki/Weibull_distribution.
* "Happiness vs income.dta1"

Multidirectional fitting?
Yes, as long as your y values stay well within the limits]0, a[.

"Weibull decay"



$$f(x) = a \cdot e^{-\left(\left(\frac{x}{b}\right)^k\right)}$$

Constraints:
a > 0 ("amplitude" of the transition);
b > 0 (determines the steepness);
k > 0.
$x_i \geq 0$, $y_i \geq 0$

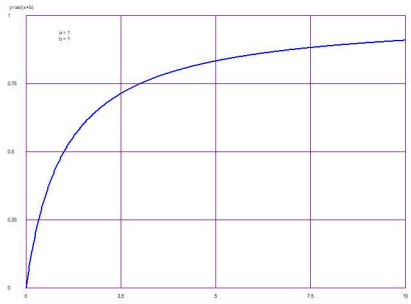
Features:
Curve descending from y = a to the horizontal asymptote y = 0.
Inflection point exists when k > 1.

Guess:
See "Weibull growth".

Examples:
* Interpreted as a cumulative distribution, it can describe the % or number of still alive persons or still working devices vs age. If y is a percentage, set a = 1, otherwise a = total number of persons/devices. See the file "Survival chances Men Belgium 2020.dta1".

Multidirectional fitting?
Yes, as long as your y values stay well within the limits]0, a[.

“Rational 1”



$$f(x) = \frac{ax}{x+b}$$

Constraints:
$a > 0$, $b > 0$, $x_i \geq 0$, $y_i \geq 0$.

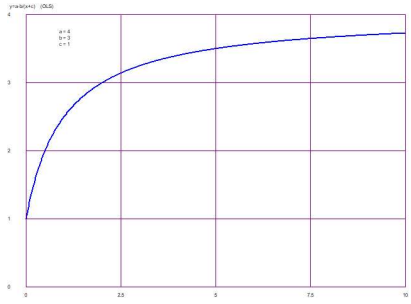
Features:
Curve through the origin; horizontal asymptote y = a.

Guess: calculated from 2 points.
All parameters can be fixed (deactivated) before the guess.

Examples:
* x = load resistance (R_L), y = voltage (U_L), a = maximal battery voltage (U_B, without load resistance), b = the desired internal battery resistance (R_B).
  See file “Internal_resistance_9V_battery.dta1”.
* Hardening of concrete?
  See file “Concrete compressive strength vs age - no additives - 192 water.dta1”.
* Michaelis-Menten kinetics: y = reaction rate (v), x = substrate concentration ([S]), b = Michaelis constant (K_M), see:
  https://en.wikipedia.org/wiki/Michaelis%E2%80%93Menten_kinetics

Multidirectional fitting?
This function is invertible as long as the y values are within]0, a[. See the remark in the “plateau-exponential” model.

“Rational 2”



$$f(x) = a - \frac{b}{x+c}$$

Constraints:
$c \geq 0$, $x_i \geq 0$

Features:
Curve not through origin; intercept: y = a-b/c;
horizontal asymptote y = a.

Guess: calculated from 3 points.
All parameters can be fixed (deactivated) before the guess.

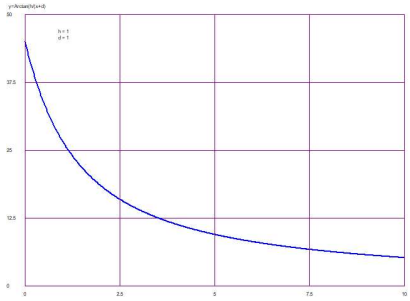
Examples:
* x = health expenditure per capita, y = life expectancy. (Approximation only.) Parameter a is the theoretical limit if an infinite amount of money is spent on health care;

a-b/c is the theoretical life expectancy if no health care was available. Extrapolating this model on the left side makes no sense though.
See file: "Cereal-crop-yield-vs-fertilizer-per-country.dta1".

* x = wine price, y = rating; see the file: "Wine ratings vs price Bordeaux 2018 RP.dta1".

Multidirectional fitting?
This function is invertible as long as the y values are all above (b<0) or all below (b>0) the limit a. See the remark in the "plateau-exponential" model.

"Parallax"



$$f(x) = \text{Arctan}\left(\frac{h}{x+d}\right)$$

Constraints:
$h > 0$, $d > 0$. $x_i \geq 0$, $y_i \geq 0$

Features:
Descending
Horizontal asymptote: y = 0.
Vertical asymptote: none if the constraints are respected.

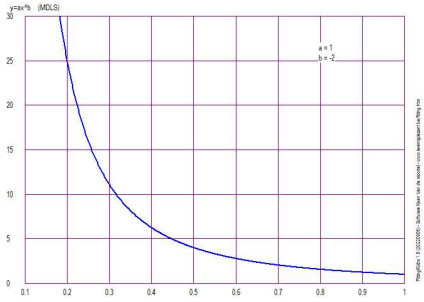
Guess: calculated from the first and the last point.
All parameters can be fixed (deactivated) before the guess.

Example:

* x+d = distance from an object (building, tree, mountain,...), d = (known or unknown) distance from the object at the nearest observation point, y = observation angle in degrees, h = the desired height of the object. See file: "Height_of_gutter.dta1".

Multidirectional fitting?
Yes, use it!

"Power"



In case the exponent (b) is *negative*, this kind of function does have a horizontal asymptote: y = 0 and a vertical one: x = 0.
See above.

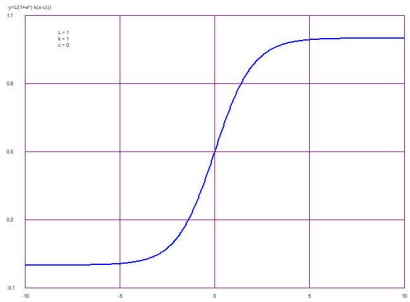
Examples:

* If x = mass of an animal, and y its heart beat, b<0.
  Parameter a will vary according to the kind of animal (mammal, bird,...).
* Volume vs pressure in a gas at constant temperature; see "Boyle.dta1"

4) Nonpolynomial monotonous functions - 2 horizontal asymptotes

"Logistic with baseline"

$$f(x) = \frac{a}{1 + e^{-\kappa(x-c)}} + b$$



Constraints:
a > 0 ("amplitude" of the transition); b ("baseline") can be anything; κ can be anything, but nothing interesting happens when it is 0.

Derived parameter:
* Upper limit ±∞: L = b+a.

Features:
Note: this is in fact a linear transformation of the "cosinus hyperbolicus" function (cosh).
Often y varies between 0 and 1 (no/yes answer); in that case, keep these parameters fixed: a = 1 and b = 0.
Horizontal asymptotes: y = b, y = b+a.
Inflection point at x = c.
Steepest slope = κ·a (in inflection point).

Guess:
$b \approx y_{min}$; $a \approx y_{max} - y_{min}$; κ and c are calculated from 2 points.
All parameters can be fixed (deactivated) before the guess.

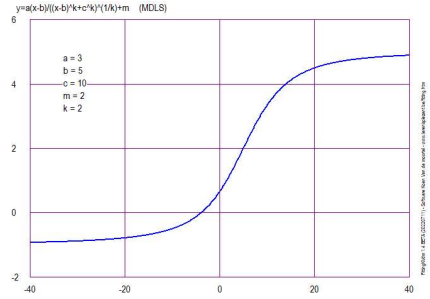
Examples:
* Transitions: x = time, y goes from b to b+a (if κ > 0) or from b+a to b (κ < 0), c = inflection point; y can be a concentration, the abundance of a word in a language, an opinion (e.g. percentage of people against slavery), the evolution in usage of mobile phones (see file "Mobile phone usage.dta1"),...
* "Tone mapping" in photography: x = brightness of light falling on a pixel, y = pixel value of red, greed and blue in a JPG image, a = 256, b = 0. High κ = high contrast.
  See example file: "JPG-Red_vs_coffee_concentration.dta1".
* x = health expenditure per capita, y = life expectancy (years) (approximation only!); see also "Rational 2".
* x = age; y = chance of survival (approximation only!). Set a fixed at 1 (100%) and b at 0.
  See: file "Survival_friends-family.dta1".

Multidirectional fitting?
This function is invertible as long as the y values are within]b, b+a[. So, use it if you see that y values are not too close to the asymptotic values (if the graph stays absolutely horizontal in the left or right area).
And at least parameter a or b has to be fixed, since it's impossible to keep both asymptotes away from the data cloud at the same time.

More information:
https://en.wikipedia.org/wiki/Logistic_function

"Transition"

$$f(x) = \frac{a \cdot (x-b)}{\left| |x-b|^k + c^k \right|^{\frac{1}{k}}} + m$$

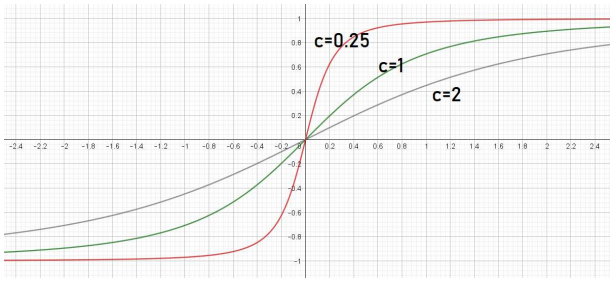


Constraints:
c > 0, k > 0

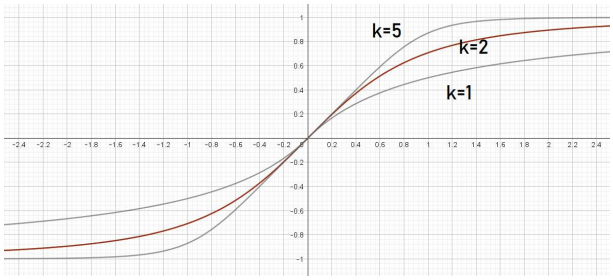
Derived parameters:
* Limit at $+\infty$: $L_1 = m+a$
* Limit at $-\infty$: $L_0 = m-a$
* Slope at inflection point: $p = a/c$

Features:
This is a generalization of the sigmoid-shaped

$$f(x) = \frac{x}{\sqrt{x^2+1}}$$



a=1, b=0, m=0, k=2, different c values



a=1, b=0, m=0, c=1, different k values

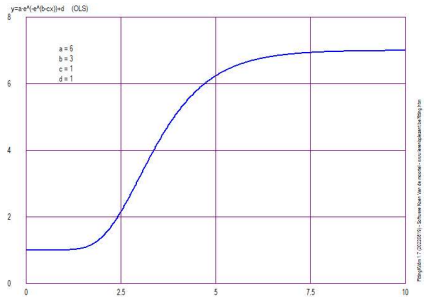
With a < 0, the curve goes from the high to the low limit.
If you want the curve to go through the origin, set b = m = 0.

Guess:
$m \approx (y_{max}+y_{min})/2$; $a \approx y_{max}-y_{min}$ (probably too high but better for convergence);
$b \approx$ average x value; c is calculated from the first and the last point; k = 2.
All parameters can be fixed (deactivated) before the guess. In most cases it's recommended to leave k fixed to a typical value (e.g. 2).

Examples:
* Usually: x = time, y = some state variable. See: "Logistic + baseline" model, which looks similar, but the curvature is different, so you could try both to find out which model fits best. Titration curves (esp. strong acid + strong base) can fit quite well with this model (fix k at a small value like 0.1), but not with the logistic one.
* See the example file "Vocabulary vs word count - De Maupassant-Short stories p1-80 & all.dta1".
* Very often, this function is "mis-used" to fit with binary data (y = 0 or 1); see: file "Survival_friends-family.dta1".

Multidirectional fitting?
Yes, as long as your y values stay well within the limits]m-a, m+a[.
And at least parameter a or m has to be fixed!

"Gompertz growth"



$$f(x) = a \cdot e^{-e^{b-cx}} + d$$

(e=Euler's constant ≈ 2.71)

Derived parameters:
* Inflection point: x_ = b/c
* Slope at inflection point: p = ac/e

Constraints:
a > 0; the start value of c can be set to 0, but it can't be fixed to 0 because the graph will stay flat.

Features:
Horizontal asymptotes: y = d, y = d+a. Usually d = 0.
Ascending if c > 0, descending if c < 0.

Guess:
$d \approx y_{min}$; $a \approx y_{max}-y_{min}$;
A quick scan gives an estimation of the steepest slope (≈ p) and its x position (≈ x_), so $c \approx pe/a$; $d \approx c \cdot x\_$.
All parameters can be fixed (deactivated) before the guess.

Examples:
* This is another sigmoid curve like the "logistic" and "transition" curves, but it tends to fit better with data from population growth or disease spreading, in cases where the growth itself is changing monotonously, like: x = age; y = chance of survival. If known, set parameter a fixed at 1 (100%) or the starting number, and d at 0.
  See file: "Survival chances Men Belgium 2020.dta1".
* Tumor growth; see file "Tumor growth Daskalakis 101.dta1".
* Ionic conductivity vs temperature, see "Pure water conductivity vs T.dta1".

Multidirectional fitting?
Yes, as long as your y values stay well within the limits]d, d+a[.
And at least parameter a or d has to be fixed, since it's impossible to keep both asymptotes away from the data cloud at the same time.

Other parametrisation:

$$f(x) = \alpha \cdot e^{-\beta \cdot e^{-\gamma x}} + \delta$$

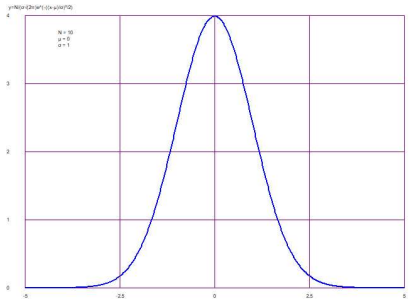
$\alpha = a$, $\beta = e^b$, $\gamma = c$, $\delta = d$.

More information:
https://en.wikipedia.org/wiki/Gompertz_function

5) Peak functions

"Gauss distribution"

$$f(x) = \frac{N}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Constraints:
$N > 0$, $\sigma > 0$; $y_i \geq 0$.

Features:
Probability density of the "normal" distribution.
x_i = center of the i'th class of quantity x; y_i = number in that class.
Top at $x = \mu$, integral (surface below the peak) = N; the bigger σ, the wider the peak.
Baseline: $y = 0$.

Guess:

$$N = \sum y_i \cdot \Delta x$$

with Δx = class (bin) width, normally $(x_{max}-x_{min})/n_{bins}$. If there are some points missing, this is not correct, but the iteration will fix this.

$$\mu = \frac{\sum x_i y_i}{\sum y_i}$$

But, another way of estimating μ is by finding the top of the peak. If the distribution is symmetric, both should be approximately the same, but the peak position is not influenced by outliers, so we take the arithmetic average of both.

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2 \cdot y_i}{\sum y_i}}$$

(= the usual standard deviation)
Another attempt of estimating σ is done by estimating the width of the peak. This is often better if there are outliers.
Then, the geometric mean of both attempts is made.
All parameters can be fixed (deactivated) before the guess.

Example:

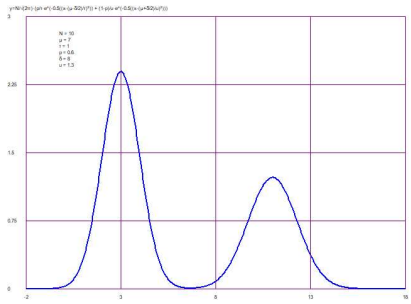
* "Normal" distribution of a variable x in a "homogeneous" group (variations in x values are purely "accidental", not systematic); x can be anything like height, mass or shoe size among people of the same age, gender, race; or any characteristic of a serially produced object,...; y = number or percentage of occurrences. N should fit to the total number of occurrences or 100% if the distribution is really normal. Parameters μ and σ are the mean value and standard deviation of the sample.
  See: https://en.wikipedia.org/wiki/Gaussian_function
* See the file "Height_boys_15.dta1".

Remarks:
Usually the x values are measurements grouped in classes, e.g. ages 20-30, 31-40, etc. Ideally, the number of classes should be approximately √N. If you have 100 measurements, group them in 10 classes.

Multidirectional fitting?
No; 0 or 1 or 2 x values are possible for each y value!

"Added Gauss distributions"

$$f(x) = \frac{N}{\sqrt{2\pi}}\left(\frac{p}{\tau}\cdot e^{-\frac{1}{2}\left(\frac{x-\left(\mu-\frac{\delta}{2}\right)}{\tau}\right)^2} + \frac{1-p}{\upsilon}\cdot e^{-\frac{1}{2}\left(\frac{x-\left(\mu+\frac{\delta}{2}\right)}{\upsilon}\right)^2} \right)$$



Constraints:
$0 < p < 1$, $N > 0$, $\tau > 0$, $\delta > 0$, $\upsilon > 0$; $y_i \geq 0$.

Derived parameters:
* # in group 1: $N_1 = p \cdot N$
* # in group 2: $N_2 = (1-p) \cdot N$
* Average x in group 1: $\mu_1 = \mu - \delta/2$
* Average x in group 2: $\mu_2 = \mu + \delta/2$

Features:
Two Gauss peaks added, clearly distinguishable if $\delta > \tau$ and υ, but they can overlap seriously if δ is small!
Baseline: y = 0.

Guess:
N, μ, σ are estimated as if it were a single Gauss peak; then, if we assume (to start) that both peaks have approximately the same height and width ($\tau = \upsilon$ and p = 0.5), it can easily be calculated that:

$$\delta = \sqrt{\sigma^2 - \tau^2} \approx \sigma$$

if we assume $\tau = \upsilon \ll \sigma$, say $\tau = \upsilon = \sigma/10$. This is often too small, but for the convergence it's better to start with too small than too big values.
Then, μ is shifted a bit left and right to find a better start value, because the S vs μ curve can have a strange shape (not like a parabola and thus difficult for the iteration method).
All parameters can be fixed (deactivated) before the guess.

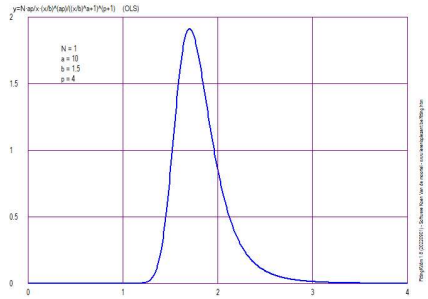
Example:
* Mixture of 2 populations; e.g. x = height of 12 year old children; μ_1 and μ_2 will be the average values of the boys and the girls, τ and υ their standard deviations. See the files "Height_distribution_adults_USA.dta1" and "Height_boys_12_18.dta1".

Multidirectional fitting?
No; 0..4 x values are possible for each y value!

Remarks:
Why did I put this p and δ in the equation and not N_1, N_2, μ_1 and μ_2 directly? Simply because N and μ are quite easy to find. That implicates that they fit in a stable way, and p and δ will converge more easily.

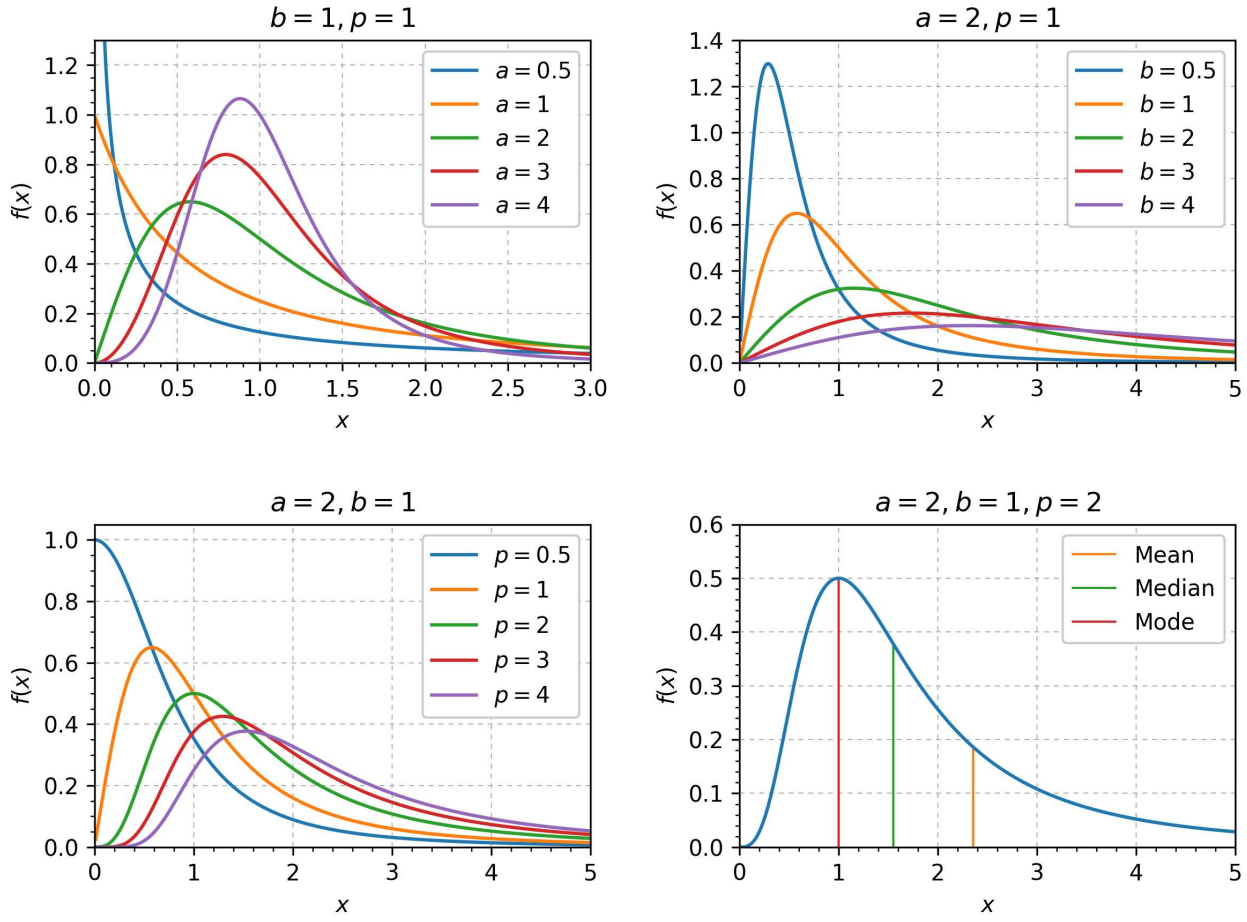
“Dagum distribution”

$$f(x) = N \cdot \frac{ap}{x} \cdot \frac{\left(\frac{x}{b}\right)^{ap}}{\left(\left(\frac{x}{b}\right)^{a} + 1\right)^{p+1}}$$



Constraints:
a, b, p, N > 0, x_i > 0, y_i > 0

Features:
Depends on the parameter values (image from Wikipedia):



See: en.wikipedia.org/wiki/Dagum_distribution

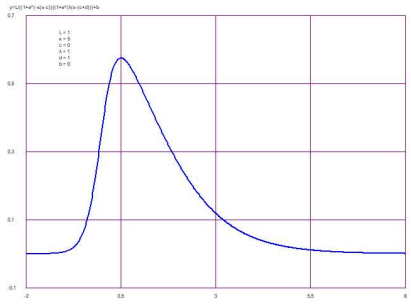
Guess:
The parameters of this distribution are very hard to estimate and interpret.
N ≈ sum of the y values · class width. If you start with a = p = 2, then b ≈ peak position.
All parameters can be fixed (deactivated) before the guess.

Example:
* This function is often used to describe income distributions, but that is not a gospel. See files: “Income distribution Belgium 2019.dta1” and “Income distribution of households USA 2020.dta1”.

Multidirectional fitting? No

“Double Logistic + baseline”

$$f(x) = \frac{a}{\left(1+e^{-\kappa(x-c)}\right)\left(1+e^{\lambda(x-(c+d))}\right)} + b$$



Constraints:
a > 0, κ > 0, λ > 0, d>0; theoretically: $y_i \geq b$, but the y values may go a bit below b due to noise.

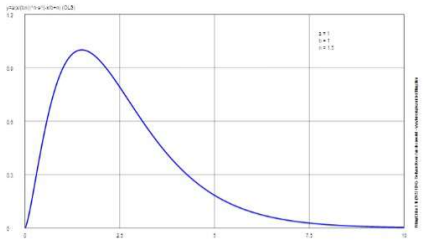
Features:
Skewed peak, κ big: steeply ascending; λ big: steeply descending.
Baseline: y = b.

Guess:
$b \approx y_{min}$; $a \approx y_{max} - b$; κ, λ, c, d: calculated from 4 points and estimated inflection points.
All parameters can be fixed (deactivated) before the guess.

Example:
* What you see when you drop a hot object in a cold liquid: x = time, y = temperature of the liquid. κ > 0 reflects the heat conductance of the liquid, λ > 0 reflects the heat loss (mostly by radiation); b = temperature of the environment, c and c+d are (more or less) the central times of the up and down transitions.

Multidirectional fitting?
No; 0 or 1 or 2 x values are possible for each y value!

“Power•Exp.decay”



$$f(x) = a \cdot \left(\frac{x}{bn}\right)^{n} \cdot e^{-\frac{x}{b}+n}$$

Constraints:
a,b,n > 0, x>0.

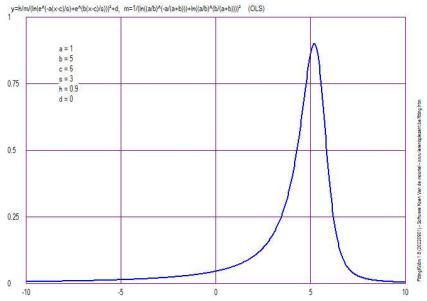
Features:
The formula looks a bit frightening, but it is designed to be easily understood: the peak point is (bn, a). The curve starts to go up from the origin, vertically if 0<n<1, linearly if n=1 and flat if n>1. Parameters a and b are just vertical and horizontal stretch factors.
In the limit for n going to 0 f becomes a simple descending exponential function.
Remark: this is similar to the Gamma probability distribution function, but differently parametrized and with n limited to positive values in order to always have a top.

Guess:
$a \approx y_{max}$; n = 1 works fine to start; then $b \approx x_{top}$.
All parameters can be fixed (deactivated) before the guess.

Examples:
* Files: “Concrete-compressive-strength vs water-cement ratio age28 no add.dta1”, “Running speeds ultra-marathons women.dta1”.

Multidirectional fitting?
No; 0 or 1 or 2 x values are possible for each y value!

“Skewed peak 1 + baseline”



$$f(x) = \frac{h}{m} \cdot \frac{1}{\left(\ln\left(e^{-a\cdot\frac{x-c}{s}} + e^{b\cdot\frac{x-c}{s}}\right)\right)^2} + d$$

with

$$m = \frac{1}{\left(\ln\left(\left(\frac{a}{b}\right)^{-\frac{a}{a+b}} + \left(\frac{a}{b}\right)^{\frac{b}{a+b}}\right)\right)^2}$$

Constraints:
a, b > 0

Features:
This may seem very complicated, but if you follow the recipe how it is brewed, it’s not.
The base is this simple V shaped function with 2 different oblique asymptotes y = -ax and y = bx:

$$g(x) = \ln\left(e^{-ax} + e^{bx}\right)$$

Using a simple transformation, we obtain an asymmetrical (if $a \neq b$) bell curve:

$$f_0(x) = \frac{1}{(g(x))^2}$$

And to make it more versatile, we add the usual scaling and shifting parameters:

$$f(x) = \frac{h}{m} f_0\left(\frac{x-c}{s}\right) + d \quad \wedge \quad m = f_0(x_m)$$

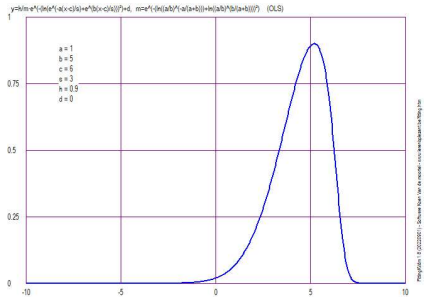
Theoretically, m is not necessary, but the height of f_0 varies a lot when a and b change, which disturbes the iteration seriously, and dividing by this height, makes h exactly the height of the peak in f.
There is actually one parameter too much, so normally you should set s=1 fixed, but if you want to force symmetry, you can fix a=b=1 and leave s free.
This function might be interpreted as a probability distribution, since its integral is finite, but it can only be calculated numerically.

Guess:
d ≈ lowest y value; h ≈ highest - lowest y value; c ≈ x with the highest y value; a = b = 1;
s ≈ estimated peak width * (a+b)/2/0.8.
All parameters can be fixed (deactivated) before the guess.

Examples:

* The distribution of ping response times can be approached by this function; see the file “ping7.dta1”.

“Skewed peak 2 + baseline”



$$f(x) = \frac{h}{m} \cdot e^{-\left(\ln\left(e^{-a\cdot\frac{x-c}{s}} + e^{b\cdot\frac{x-c}{s}}\right)\right)^2} + d$$

with

$$m = e^{-\left(\ln\left(\left(\frac{a}{b}\right)^{-\frac{a}{a+b}} + \left(\frac{a}{b}\right)^{\frac{b}{a+b}}\right)\right)^2}$$

Features:
To understand this function, start from the same base function g (see skewed peak 1), but then do this transformation to obtain a bell curve:

$$f_0(x) = e^{-(g(x))^2}$$

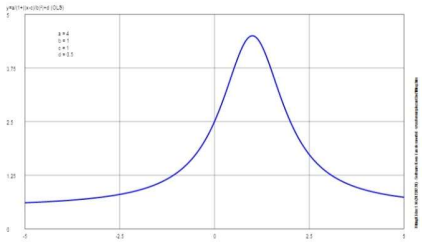
and then calculate m and add the scaling parameters as with peak version 1.

Guess:
See peak version 1.

Examples:
* See peak version 1.
* See the file “Hot_stone_water.dta1”.

“Lorentzian peak + baseline”

$$f(x) = \frac{a}{1 + \left(\frac{x - c}{b}\right)^2} + d$$



Features:
a peak centered around x = c, amplitude a, baseline y = d; b determines the width.

Constraints:
b > 0

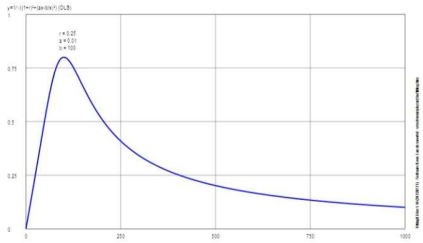
Guess:
c ≈ x position of y_{min} or y_{max}; d ≈ y_{max} or y_{min}; a ≈ y_{top} - d;
b ≈ (x_{max}-x_{min})/2 (This is not precise but normally OK to start with.)

Derived parameters:
* the integral of the peak (minus baseline): I = πab

Examples:
* Often: x = frequency, y = optical/mechanical vibration intensity… The y variable peaks when there is some kind of resonance phenomenon.

“RLC serial filter”

$$f(x) = \frac{1}{\sqrt{(1+r)^2 + \left(ax - \frac{b}{x}\right)^2}}$$



Constraints:
$a \geq 0$, $b \geq 0$, $r \geq 0$, $x > 0$, $0 < y < 1$

Features:
If $a > 0$ and $b > 0$: curve ascending from origin (= limit) to a peak, then slowly descending to 0 at infinity.
If $a = 0$ or $b = 0$ (not both), the curve is monotonous; see the special cases “RLC serial filter - preset RL” and “RLC serial filter - preset RC”.

Derived parameter:
* Peak position (resonance frequency): $x = \sqrt{(b/a)} = 1/(2\pi \sqrt{(LC)})$

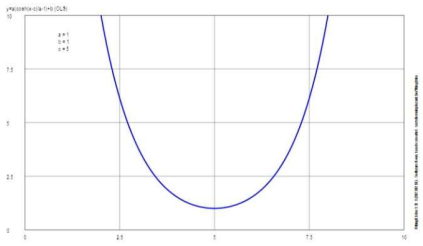
Guess:
Peak position ≈ x value with maximal y; $r \approx 1/y_{max}-1$; a and b are calculated with another point.

Example:
* x = frequency, $y = U_{out}/U_{in}$ with U_{in} the AC voltage over a coil (inductance L in henry, internal resistance R_i in ohm) + capacitor (capacitance C in farad) + resistor (resistance R in ohm) in series; U_{out} is measured over the resistor (“band pass RLC filter”). See example file: “Filter 1 RLC.dta1”
  Meaning of the fitted parameters: $r = R_i/R$; $a = 2\pi L/R$; $b = 1/(2\pi RC)$. The peak height will only reach 1 if the coil is “ideal” (zero internal resistance).

Multidirectional fitting?
Only in the special cases when $a = r = 0$ (RC filter) or $b = 0$ (LC filter) and as long as the y values are within]0, 1[.

6) Other functions with 1 extremum

“Chain line”



$$f(x) = a \cdot \left(\cosh\left(\frac{x-c}{a} \right) - 1 \right) + b$$

This is a linear transformation of the hyperbolic cosine function ($\cosh(x) = (e^{x}+e^{-x})/2$.

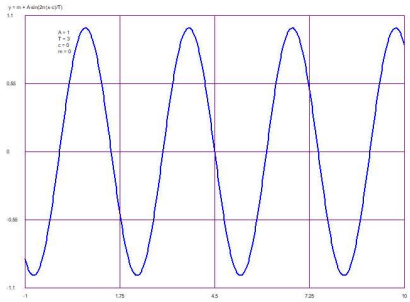
Features: This is a parabolic looking curve with top (c,b); U shaped if a > 0, and upside down if a < 0; it reduces to a flat line y = b if a = 0.

Guess:
If the points form a convex pattern, c = x value with minimal y, otherwise c = x value with maximal y;
b is the corresponding y value;
a is estimated from another point using the Taylor approximation ($\cosh(x) = 1+x^2/2+\ldots$).

Examples:
* a hanging chain or cable (see file “Chain.dta1”);
* a well constructed arch in a building (e.g. a bridge).

7) Periodic and semiperiodic functions

"Sine wave"



$$f(x) = m + A \cdot \sin\left(2\pi \frac{x - c}{T}\right)$$

Constraints:
A > 0, T > 0

Derived parameters:
* Pulsation: ω = 2π/T
* Frequency: f = 1/T. If T is in seconds, f is in Hertz.

Features:
Curve goes up and down around the average y = m, with period T, amplitude A, phase shift c. See e.g. en.wikipedia.org/wiki/Sine_wave

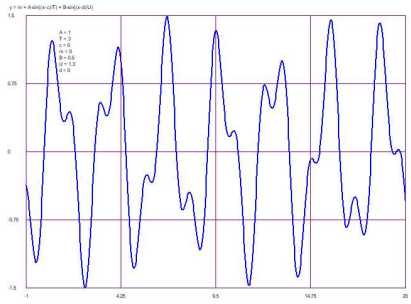
Guess:
$m \approx y_{average}$; $A \approx \sqrt{2}$·standard deviation of y values; T is estimated by crosscorrelating the series y_i-m with $f(x)=\sin(2\pi x/T_{test})$ with T_{test} varying over a range (from $3(x_n-x_1)/n$ to $(x_n-x_1)/2$), then c is estimated by crosscorrelating with $f(x)=\sin(2\pi(x-c_{test})/T)$ with c_{test} varying from -T/2 to T/2.
Parameters that can be fixed (deactivated) before the guess: a, A, c, T.

Example:
* Any periodic variable, so x = time, y might be the average daily outside temperature (or the daily maximum or minimum) on a certain place, T = 1 year, m = average temperature, c = time of the year when it's warming up the fastest.
  See file "Temperature Chatanga.dta1".
* But x (and T) can also be spatial distance, then y could be a height, intensity of a standing wave,...

Multidirectional fitting?
No; 0 or an infinite number of x values are possible for each y value!

Remarks:
* The guessed parameters will usually be good if you have sufficient data, covering at least one period.
* Do enough measurements, and do them at irregular time intervals, in order to minimize the risk to find false frequencies ("aliasing effect", see: https://en.wikipedia.org/wiki/Aliasing).
* If T is known, use it! E.g. if t is the time in days, and you know that your periodicity is 1 year, enter T = 365.25 and set "active" to false (uncheck).
* For all the periodic models, it's best to reduce the x values to smaller numbers. E.g. if x = time in years, like 2000, 2001, 2002 etc., change it to 0, 1, 2, etc., so x = years since 2000. This improves the stability of the period estimation.

"Added sine waves"



$$f(x) = m + A \cdot \sin\left(2\pi \frac{x-c}{T}\right) + B \cdot \sin\left(2\pi \frac{x-d}{U}\right)$$

Constraints:
A > 0, T > 0, B > 0, U > 0

Derived parameters:
* Pulsation 1: $\omega_1 = 2p/T$
* Frequency 1: $f_1 = 1/T$
* Pulsation 2: $\omega_2 = 2p/U$
* Frequency 2: $f_2 = 1/U$

Features:
Sum of two sine waves, so only strictly periodical if T and U are integers (the period is then least common multiple of T and U).

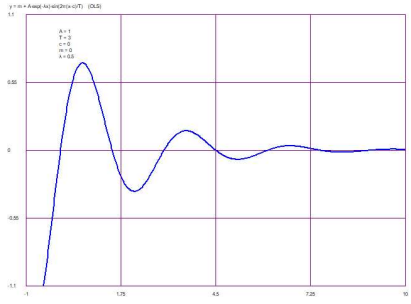
Guess:
m, A, T, c: see "Sine wave"; the second period (U) is just set a bit smaller than T, and the second amplitude (B) 10x smaller than A to start with. Since it is not easy to estimate U, it's best that you enter a reasonably expected value before iterating.
Parameters that can be fixed (deactivated) before the guess: a, A, c, T, λ.

Examples:
* x = time, y = tidal height or current, T = 12h, U = 12.4206h (Sun and Moon rhythms). See file "Tidal_current_Fundy.dta1".
* x = time, y = temperature, T = 365.25 days, U = 1 day (yearly and daily rhythms). This is very difficult to fit because the "noise" is often bigger than the amplitude B.

Multidirectional fitting?
No; 0 or an infinite number of x values are possible for each y value!

Remarks:
Clicking the "Guess" button will usually produce a good result for T, but U is much more difficult to find. So, if you have a good idea about the periods, enter your guesses and lower and upper limits!

"Damped sine wave"



$$f(x) = m + A \cdot e^{-\lambda x} \cdot \sin\left(2\pi \frac{x-c}{T}\right)$$

Constraints:
A > 0, T > 0, λ > 0

Derived parameters:
* Pulsation: $\omega = 2p/T$
* Frequency: $f = 1/T$

Features:
Not strictly periodic (unless λ = 0), but we could describe it as a periodic curve with descending amplitude (the bigger λ, the quicker it descends).

Guess:
m, A, T, c: see "Sine wave"; λ is estimated by comparing the averages of $|y_i|$ of the first and the second half of the points.
All parameters that be fixed (deactivated) before the guess.

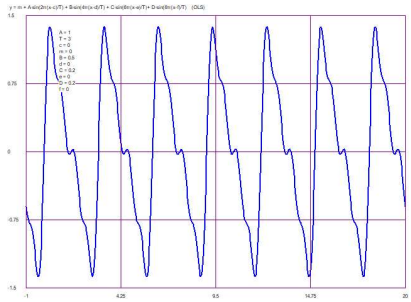
Example:
* Usually x = time, y might be the displacement of a realistic pendulum with friction. The bigger λ, the bigger the friction. See file "Pendulum.dta1".

Multidirectional fitting?
No; 0 or an infinite number of x values are possible for each y value!

"Sine wave with harmonics"

$$f(x) = m + A \cdot \sin\frac{2\pi(x-c)}{T} + B \cdot \sin\frac{4\pi(x-d)}{T} + C \cdot \sin\frac{6\pi(x-e)}{T} + D \cdot \sin\frac{8\pi(x-f)}{T}$$

Constraints:
A > 0, B > 0, C > 0, D > 0, T > 0



Derived parameters:
* Pulsation: ω = 2p/T
* Frequency: f = 1/T

Features:
The period is T, but the repeated pattern can have a complicated shape.

Guess:
m, A, T, c: see "Sine wave"; all other amplitudes are set to small amplitudes and the phases are set to 0 to start with.
All parameters can be fixed (deactivated) before the guess.

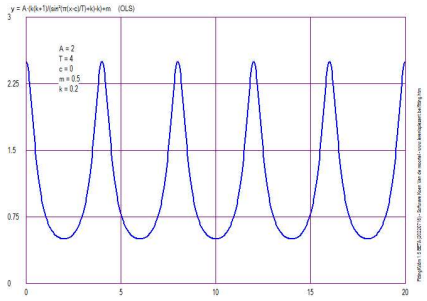
Examples:
* Usually x = time, y = any kind of quantity that varies periodically but in a more complex way than simply "wave like", a heart beat, a sound, the number of sun spots, animal populations,... See file: "Lynx.dta1"

Multidirectional fitting?
No; 0 or an infinite number of x values are possible for each y value!

For the stability of the iteration, you may want to begin with setting B, C and D fixed to zero, do some iterations and adjustments to get the main period correct, and then activate B, C, D.

“Periodic peaks”

$$f(x) = A \cdot \left(\frac{k(k+1)}{\sin^2\left(\pi \cdot \frac{x-c}{T}\right) + k} - k \right) + m$$



Constraints:
T > 0, k > 0

Features:
This may seem complicated, but it is only a generalization of:

$$f_0(x) = \frac{1}{\sin^2 x + 1}$$

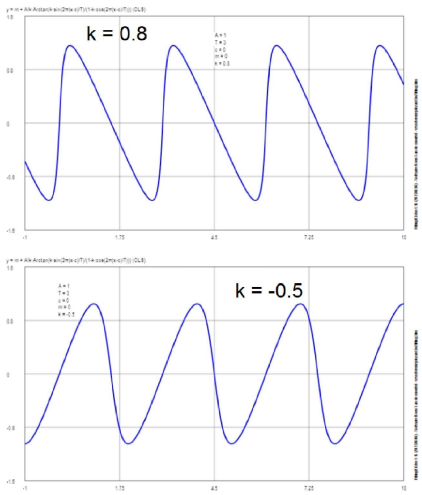
so that the baseline is y = m, the peak height A, and the period T. The smaller k, the sharper the peaks.

Guess:
A, T, c: see “Sine wave”; but m is set to y_{min}; k = 1.
All parameters can be fixed (deactivated) before the guess.

Multidirectional fitting?
No; 0 or an infinite number of x values are possible for each y value!

Examples:
* x = time, y = population of a predator like the lynx (see the example files).
* x = time, y = number of sunspots (oversimplified approximation);
  see “Sunspots-monthly avg.dta1”.

“Skewed wave”



$$f(x) = m + \frac{A}{k} \cdot \mathrm{Arctan}\left(\frac{k \cdot \sin\left(2\pi \frac{x-c}{T}\right)}{1 - k \cdot \cos\left(2\pi \frac{x-c}{T}\right)} \right)$$

Constraints:
$A > 0$, $T > 0$, $-1 \le k \le 1$

Derived parameters:
Same as “Sine wave”

Features:
This function is periodic with period T, but the peak position is not in the middle of the average crossing points. The first top is at:

$$x_{top} = T \cdot \mathrm{Arccos}(k) + c$$

And the amplitude is:

$$y_{top} - m = \frac{A}{k} \mathrm{Arctan}\left(\frac{k}{\sqrt{1-k^2}} \right)$$

With k = 0, strictly, the function is undefined, but the limit is the regular sine function.
If $k > 0$, the ascending slope is steeper than the descending slope.
If $|k| = 1$, the function becomes a “sawtooth”.
Amplitude $\approx$ A when $|k|$ is small, $\pi/2 \cdot A$ when $|k| = 1$.

Guess:
m, A, T, c: see “Sine wave”; k is set to 0 to start with.
All parameters can be fixed (deactivated) before the guess.

Examples:
* Use this model whenever you would use a sine, but suspect an asymmetry. See the files: “Temperature Chatanga.dta1” and “Temp Popovica 20230804.dta1”.

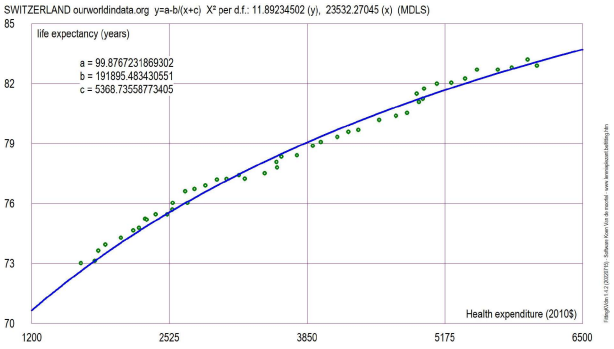
If you need another model, please let me know! I will implement it in the next version!

About “goodness of fit” and “confidence intervals”

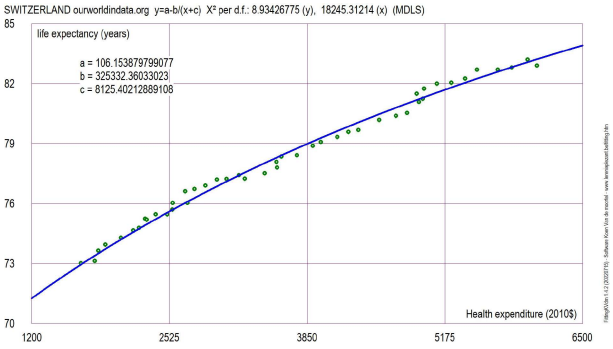
Of course, you will want to know how reliable your model and parameters are. What can we say about that?
Of course, visually inspecting the graph will tell you a lot already, but there are other means.

Speed of convergence

A good indicator of the appropriateness of the model is the number of iterations needed to obtain stable parameters. If the characteristics of the model functions (slope, extrema, periodicity, infliction points, asymptotes etc. can be detected easily in the data, only a small number of iterations (like 10..100) will be needed in order to keep the first 5 significant digits of the parameters stable.
For example: if you want to find a periodicity in your data but your points only form like the half of a wave, then there might be a whole range of periods possible. If you have points that go up and down several times, the period will be found quickly without much uncertainty.
So, if you need to keep iterating thousands or millions of times and you see the parameters change every time, while the curve seems to stay the same visually, that’s a bad sign. It simply means that you need more data to decide if the model is appropriate, and those data should be in different areas of the domain of the model function.
Example: take a look at the file “Life exp vs health spending CH.dta1”. The points are more or less on an ascending line, but you know they can’t be, since that would go below zero on the left side, and towards infinity on the right side. The curvature is not really clear enough to decide which kind of function is the best to describe it, so whatever model you try, it will converge slowly.
Compare the graphs below: the second curve (after 12000 iterations) is pretty much the same as the first one (after 3000 iterations) but the parameters changed drastically: the life expectancy went from 99.88 to 106.15 years. This simply means we can not conclude anything from this model, sorry! The logistic model is more stable in this case and hence a little bit more reliable.



After 3000 iterations



After 12000 iterations

Chi-squared

In most cases, after a number of iterations, the parameter values become more and stable, unless the data are very noisy or the model function is clearly inappropriate. The S value, will reach a relative minimum. The values of χ^2 (per degree of freedom) in the y and the x direction (if the function is invertible), related to S, are shown after each iteration:

$$\chi_y^2 = \frac{1}{n - n_{pars}} \sum_{i=1}^{n} \frac{\left(y_i - f(x_i)\right)^2}{\sigma_{y,i}^2}$$

$$\chi_x^2 = \frac{1}{n - n_{pars}} \sum_{i=1}^{n} \frac{\left(x_i - f^{-1}(y_i)\right)^2}{\sigma_{x,i}^2}$$

(n_{pars} is the number of active parameters, so $n-n_{pars}$ is the degrees of freedom number.)

Usually in other software, in the best case, only the first one is calculated.
What can be concluded from these numbers? If the residues in the vertical direction ($r_{y,i} = y_i-f(x_i)$) are in the same order of magnitude as the estimated uncertainties of the y values ($\sigma_{y,i}$), then χ_y should be around 1. If it is smaller than 1, it means the function fits better than expected, or... the $\sigma_{y,i}$ values are overestimated. If it is much bigger than 1, it can mean that the fit is bad (wrong model), or... that the $\sigma_{y,i}$ values are underestimated! Therefore, it is of great importance to enter realistic values for $\sigma_{y,i}$! The same reasoning can be done for the horizontal residues ($r_{x,i} = x_i-f^{-1}(y_i)$) of course.
So, χ^2 values around 1 are okay, less is usually better, but don't rely only on them to judge.
See also: https://en.wikipedia.org/wiki/Reduced_chi-squared_statistic.

Most regression programs calculate the so-called coefficient of determination r^2, the square of the Pearson correlation coefficient, which can be written in this form:

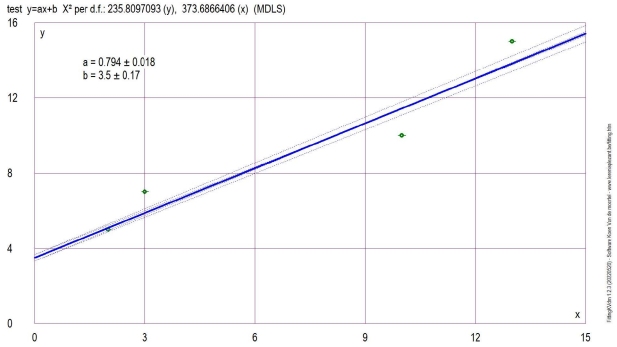
$$r^2 = 1 - \frac{\sum\left(y_i - f(x_i)\right)^2}{\sum\left(y_i - \bar{y}\right)^2}$$

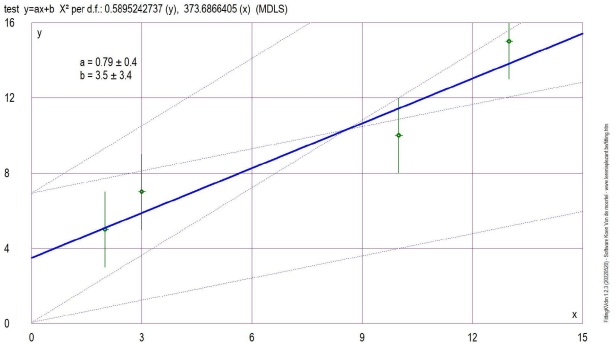
with f(x) = ax + b. It's only meant to be used for linear models, which is a big limitation.
If the point cloud looks like a cigar, r^2 is close to 1, if it looks round, r^2 is close to 0. If all y values are exactly aligned, $r^2=1$, but if that line is horizontal, a catastrophic error occurs, which is kind of strange.
See for example: https://online.stat.psu.edu/stat462/node/95/
I often notice that people think of this r^2 as an indicator of the "goodness of fit" of their model, but that is a mistake. χ^2 is much better for that. Just consider this example:
The measurements:

x	2	3	10	13
y	5	7	10	15

Suppose the x values all have a precision of ±0.1.

Now, in the left graph below, we gave the y values a precision of ±0.1; in the right one the precision was ±2.





Of course the obtained regression line (linear model, MDLS) was the same, but the uncertainty in the parameters is different (see p.55):
Left: a = 0.794 ± 0.018, b = 3.5 ± 0.17
Right: a = 0.79 ± 0.4, b = 3.5 ± 3.4

In both cases, r^2 was the same of course (0.919783, calculated with Quatto Pro), but in the vertical direction, the χ^2 values were very different!
Left: $\chi^2 \approx 236$; this means that there are certainly other variables besides x influencing y!
Right: $\chi^2 \approx 0.59$; the regression line passes through all the error flags; this means that the linear model might well be correct, but we need more precise measurements to have more certainty.
So, the conclusions that can be drawn from χ^2 are much more interesting than those we get from just looking at r^2, and χ^2 can be used for any model! You also see that a good estimation of the precision of your data is really necessary if you want to draw the right conclusion!

Use common sense!

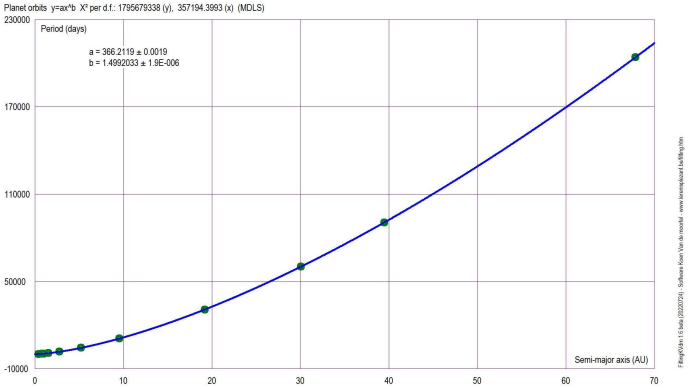
By far the best way to judge if a model is appropriate, is: check if it is supported by scientific reasoning, definitely *not* by comparing the fitting results! Just consider for example that you can always fit an n-1th degree polynomial perfectly through a set of n data points! A model with more parameters will generally "fit" better, because the curve can adapt better to the data, but that doesn't mean it is a better model!
A good model usually gives you good extrapolated values, a bad model will most certainly produce nonsense if you feed it with x values beyond the range of your dataset. E.g.: you can fit a straight line through any dataset, but ask yourself: "Does it mean anything if I extrapolate it to the point where the line goes below zero?".

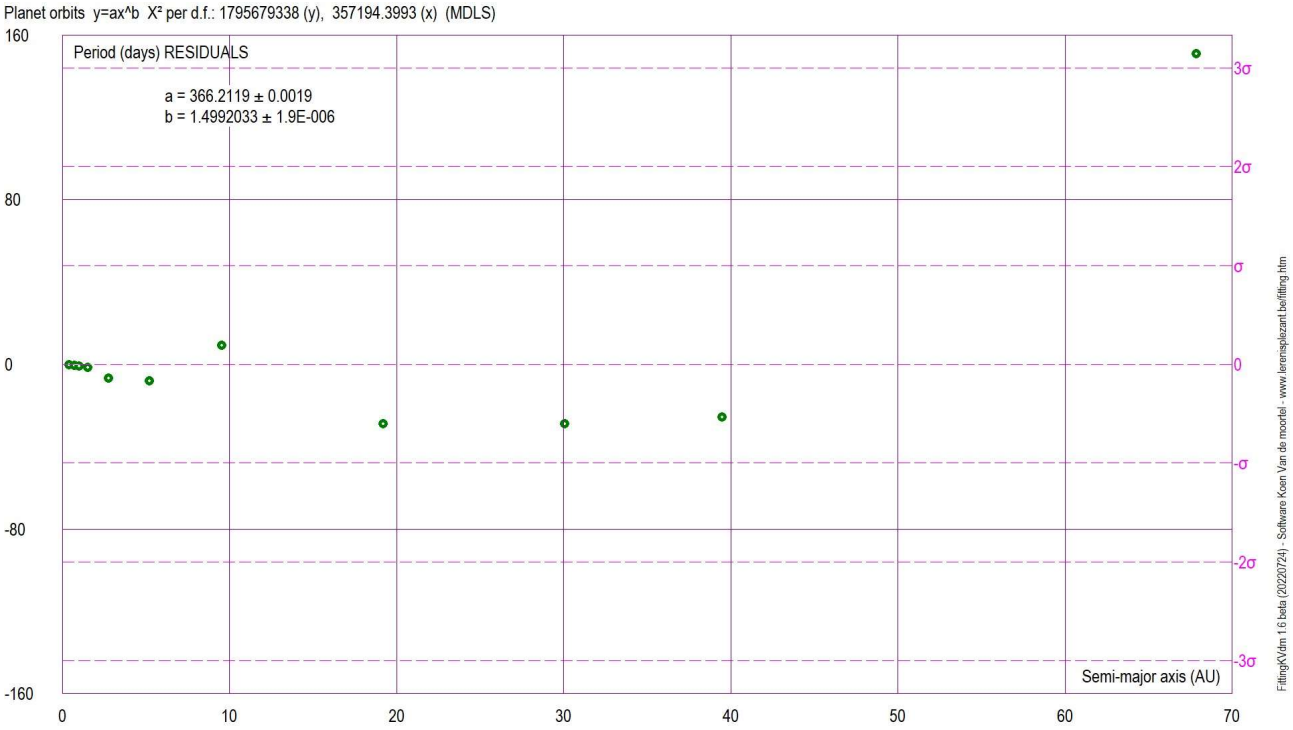
Residuals

If you doubt if the scientific reasoning explains your data sufficiently, you might click the "Show residuals" selector. If you see that the residuals are randomly spread around 0,

that's a good sign. If you recognize a clear pattern in the points, you might have to adjust your model formula.

You can also use this graph to detect so-called "outliers", data points that don't seem to "obey" the model very well. For example: if you look at the power function fitted through the dataset of the orbit times of planets and other objects vs their axis length, it looks very nice, but if you look at the residuals, you see that the last object (Eris) is more than 3 standard deviations away from the model function.



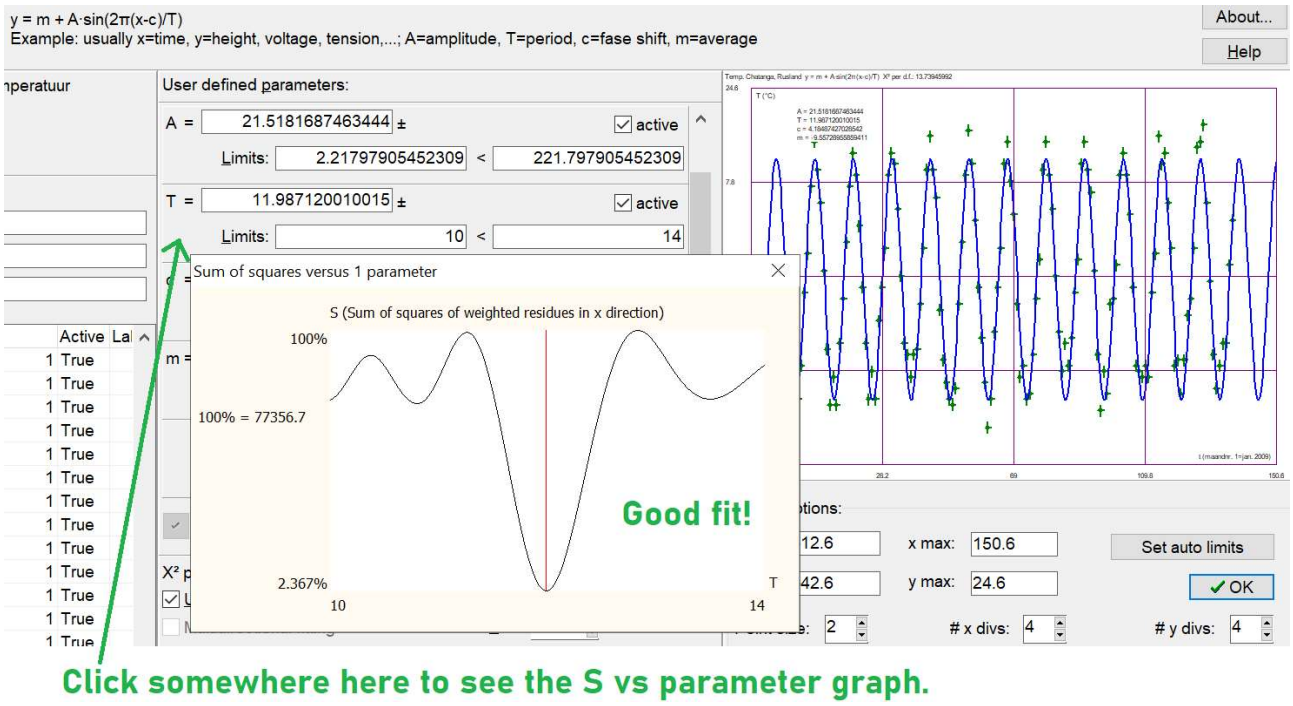


If desired, you can also click the "Save residuals" button, and they will be stored in a file with the same name, but "residuals" added.

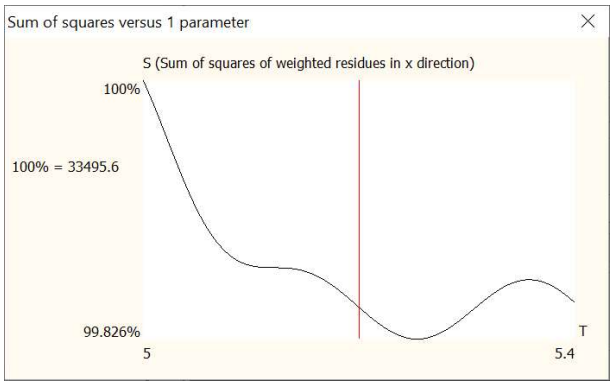
S versus parameter graph

Another tool this program offers to evaluate the "goodness of fit" for a model, is this: click somewhere in the panel of a parameter; a small graph will appear and show the relative variation of S versus the parameter. If the regression was done successfully, the actually found parameter value (indicated by a red line) should be in a clear relative minimum of this graph. Below you see the analysis of 13 years of temperatures (monthly averages) of a place called Chatanga (Siberia). The algorithm finds a period of 11.987 months, not precisely 12, but that's normal: months don't have equal lengths! It would be better in such cases to measure the time in days since a certain starting day and expect

a period of 365.25 days. Anyway, this 11.987 is obviously in a deep minimum, so the chance that this period is "pure coincidence" is very low.



On the other hand, if we would want to check if a period of, say 5.2, exists in the data, you could enter 5.2 in the box next to "T =", confirm (v button a bit below), and then ask the S vs T graph again. Then we would see only a very small dip in the S plateau and the 5.2 is not in the minimum. Of course there is no physical reason to expect a periodicity of 5.2 months.



Confidence interval estimation

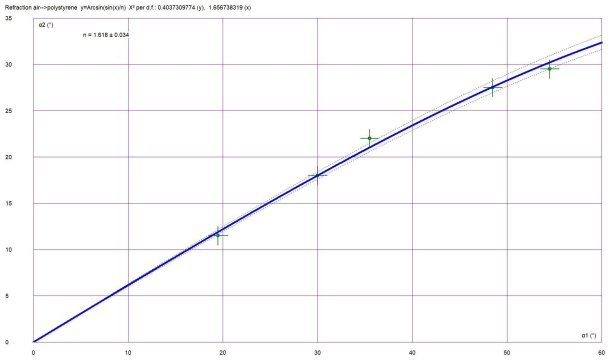
What happens when you click the button "Estimate parameter confidence intervals"?
To each x and y value, some random noise is added with a standard deviation equal to the entered σ_x and σ_y. Then a number of iterations is done (depending on the model) to produce new parameter values with these noisy data. This is done a 100 times, and the standard deviations of these values are calculated. If the σ_x and σ_y values are realistic, AND if sufficient iterations were done to get stable parameters, the so obtained standard deviations should be realistic estimations of the parameter confidence intervals, taking into account both the uncertainties in the data as well as the error propagation trough the model function.

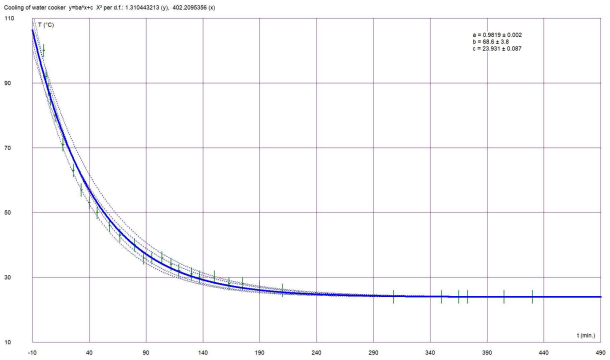
Remarks:

* If σ_x and σ_y are normal standard deviations (meaning: the chance the entered x_i value is between $x_i-\sigma_{x,i}$ and $x_i+\sigma_{x,i}$ is approximately 68%), so will be the parameter deviations. If σ_x and σ_y are 95% confidence intervals, so will be the parameter deviations.
* If you hit the button again, you might get slightly different deviations. There is no way to get them with absolute precision. It's the order of magnitude that counts.
* It makes no sense to do this calculation before the parameters came to convergence.
* With large datasets this calculations may take some time. Be patient…

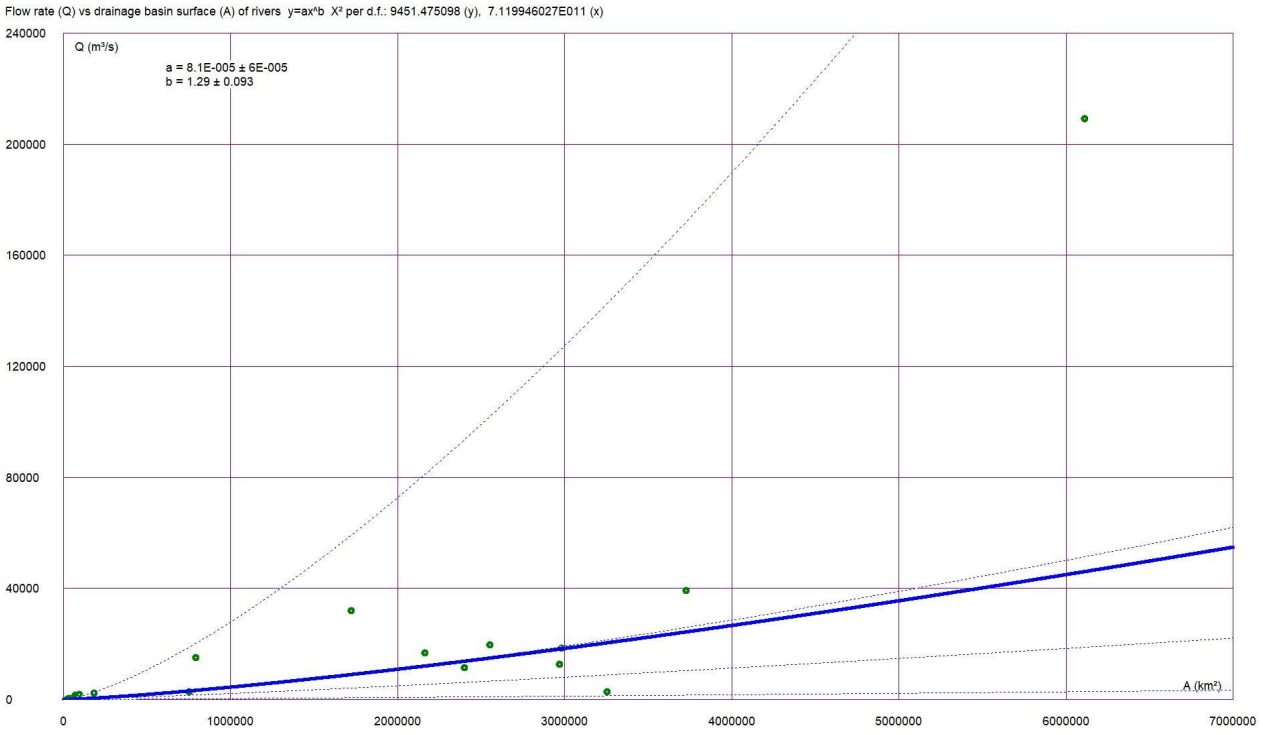
Once you did this calculation, you will see some extra dotted lines on the graph (if you checked "Show curves with deviations"). These represent "worst cases": if there is only 1 parameter p, you will see 2 dotted lines, one where p is replaced by p+σ and one with p-σ (σ being the error, or deviation if you will, on the parameter). If there are 2 parameters, you will see 4 dotted lines: (1) with $p_1-\sigma_1$ and $p_2-\sigma_2$, (2) with $p_1-\sigma_1$ and $p_2+\sigma_2$, (3) with $p_1+\sigma_1$ and $p_2-\sigma_2$, (4) with $p_1+\sigma_1$ and $p_2+\sigma_2$. With 3 parameters, you'll see 8 dotted lines, etc.
This will give you a good idea of the reliability of your model.

These examples (with 1 and 2 parameters) look quite good:





The following example is an attempt to fit the flow rate (Q) of rivers versus their drainage basin surface (A) with a power function: $Q=aA^b$. The line doesn't fit very well, but the graph can give you an idea of the average rainfall in the basin: above the line = much rain, below the line = little rainfall. BTW, the highest point is the Amazon of course.
Notice the big uncertainty in the a parameter: $a=(8.1\pm6)\cdot10^{-5}$ and the high χ^2 values: 9451 (vertical) and $7.12\cdot10^{11}$ (horizontal).


Flow rate (Q) vs drainage basin surface (A) of rivers y=ax^b X² per d.f.: 9451.475098 (y), 7.119946027E011 (x)
Q (m³/s)
a = 8.1E-005 ± 6E-005
b = 1.29 ± 0.093
240000
200000
160000
120000
80000
40000
0
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
A (km²)

Example data files (in alphabetical order)

In order to give you something to play with, I give you some example data files. You can use them for training or educational purposes.
You will find them in the folder \data\examples in the program file folder.

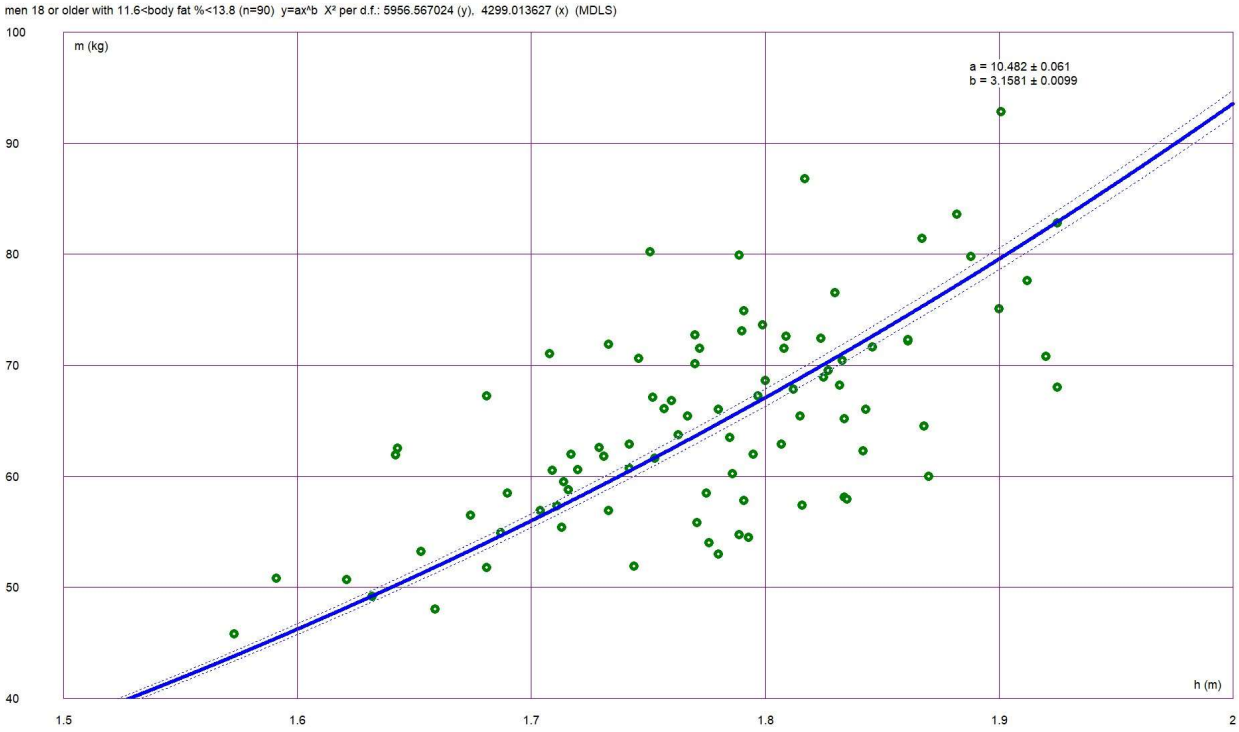
---

* **adult men 11.6-13.8 percent fat - h vs m.dta1**
* **adult men 11.6-13.8 percent fat - m vs h.dta1**

Variables: x = mass (m, in kg); y = height (h, in m) (and vice versa) of 90 adult American men with fat percentage between 11.6 and 13.8.

Source: NHANES, thanks to Nir Krakauer.

Try the "Power" model and notice how dramatically the parameters change if you choose the classical or multidirectional fitting. This shows how wrong the BMI ("body mass index") is.



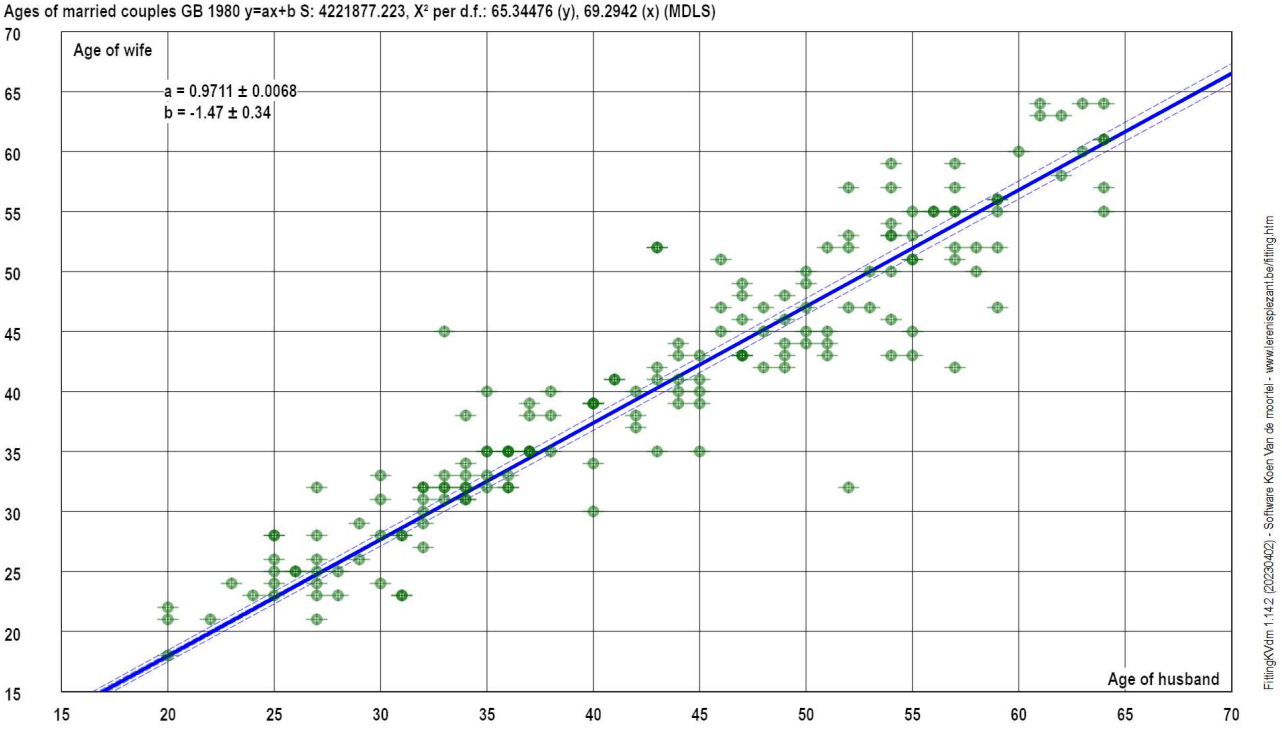
See also:
www.researchgate.net/publication/358736303_The_Body_Mass_Index_recalculated

* Ages of married couples.dta1

Variables: x = age of husband, y = age of wife (years) from 170 random married British couples in 1980.

Source: OPCS study in Great Britain, by C.Marsh (1988), quoted in "A Handbook of Small Data Sets", by D.J. Hand, F. Daly, A.D. Lunn, K.J. McConway and E. Ostrowski, Springer 1994.

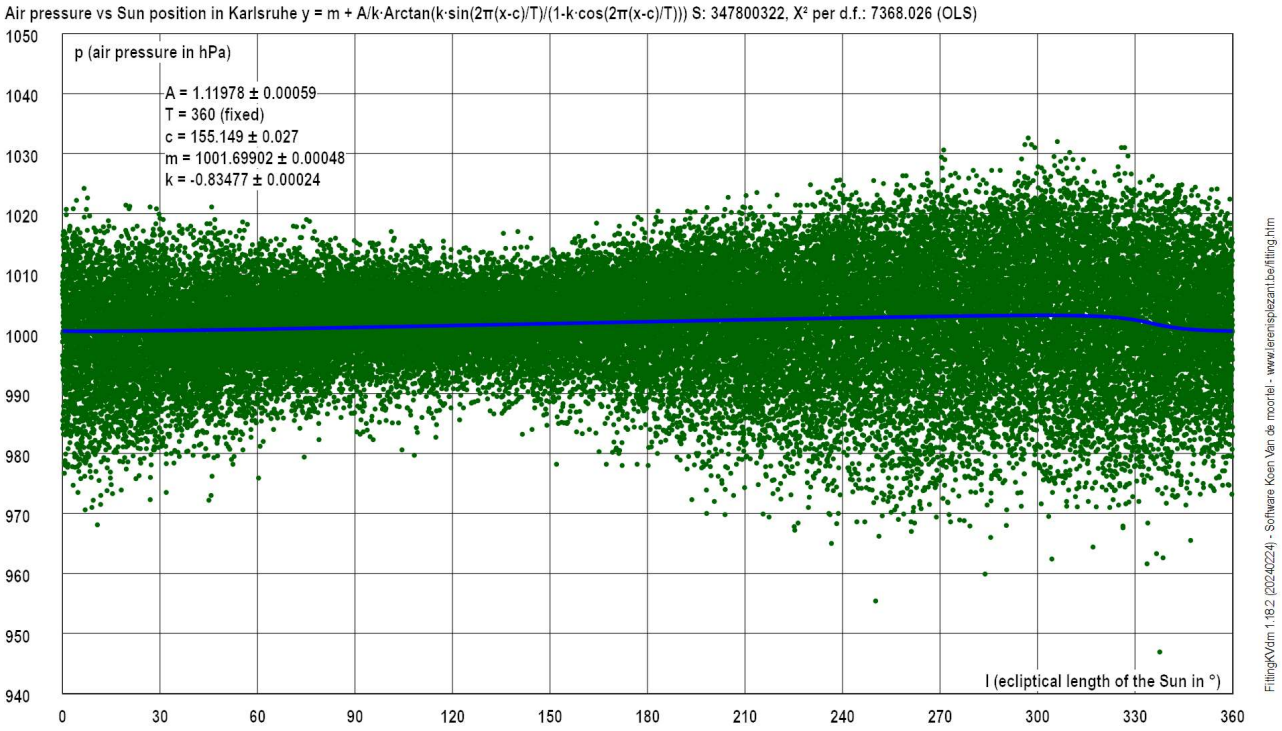
This is a typical example where you can expect a more or less linear relationship, because in most cases there is not very much age difference between spouses.



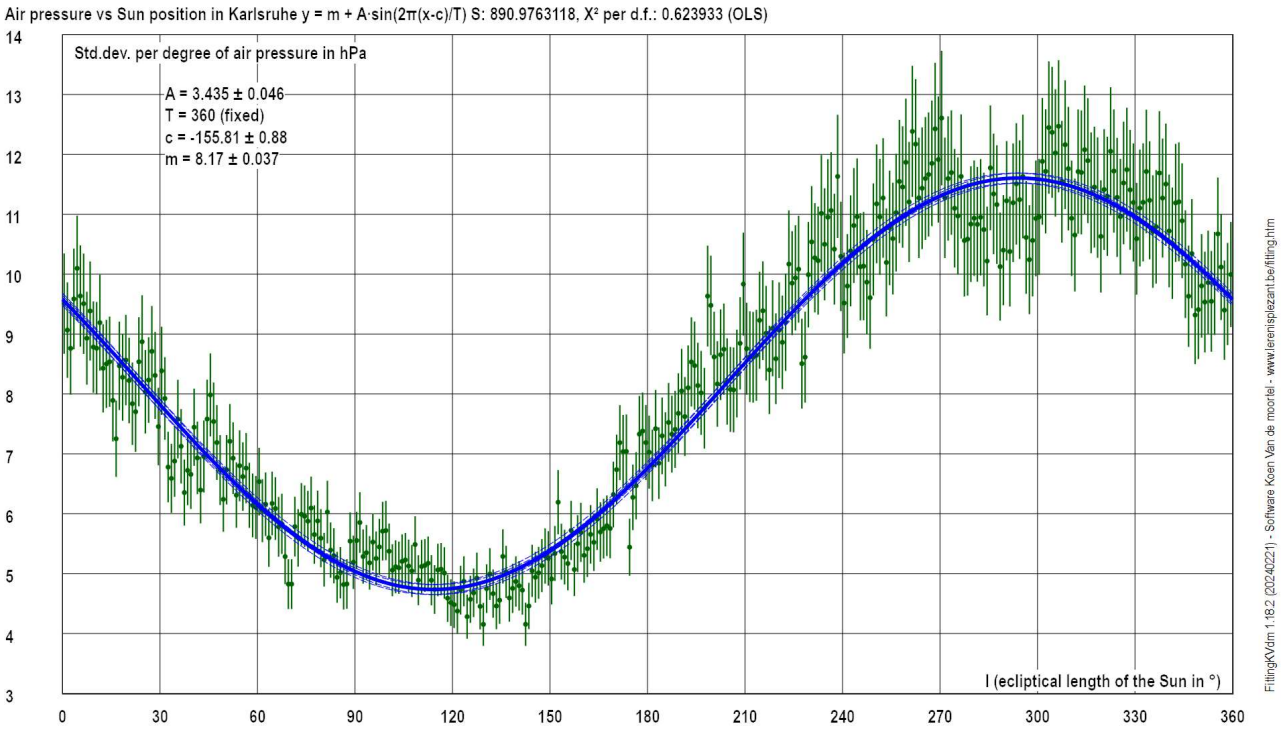
* **Air pressure vs Sun position 0-360 in Karlsruhe.dta1**
* **Air pressure vs Sun position in Karlsruhe.dta1**
* **Air pressure vs Sun position in Karlsruhe SD per degree.dta1**

Variables: x = Sun position in ecliptical longitude (0..360° in the first file, cumulative in the second file), y = air pressure in hPa. Third file: x = Sun position interval [x-0.5, x+0.5[center, and y = standard deviation of the pressure for each degree.
The measurements were done every morning 1 Jan. 1876 to 1 March 2006, except between Nov. 1944 and Sep. 1945 in Karlsruhe, Germany.
Source: www.dwd.de/EN/ourservices/_functions/search/search_Formular.html

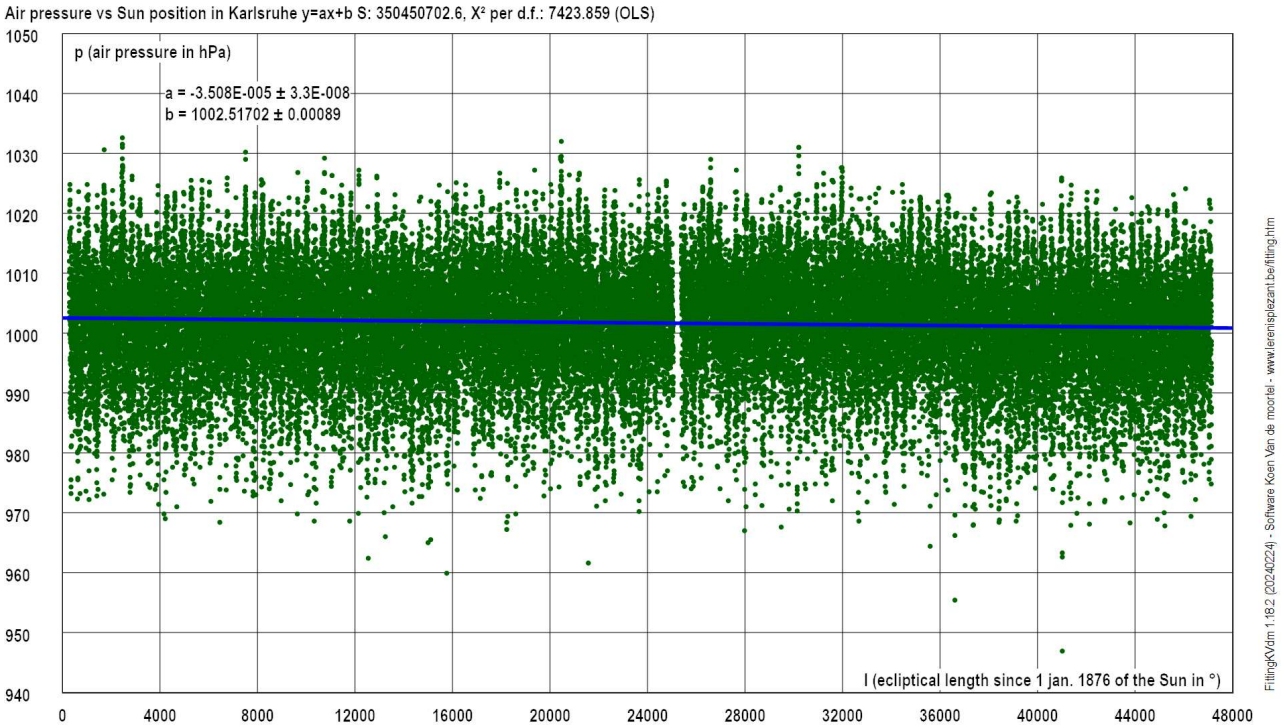
The average pressure is almost constant throughout the year; a skewed wave with a very small amplitude fitted looks like this:



The variability (standard deviation) is clearly seasonal though:



And there is a very tiny downward trend in all those years:

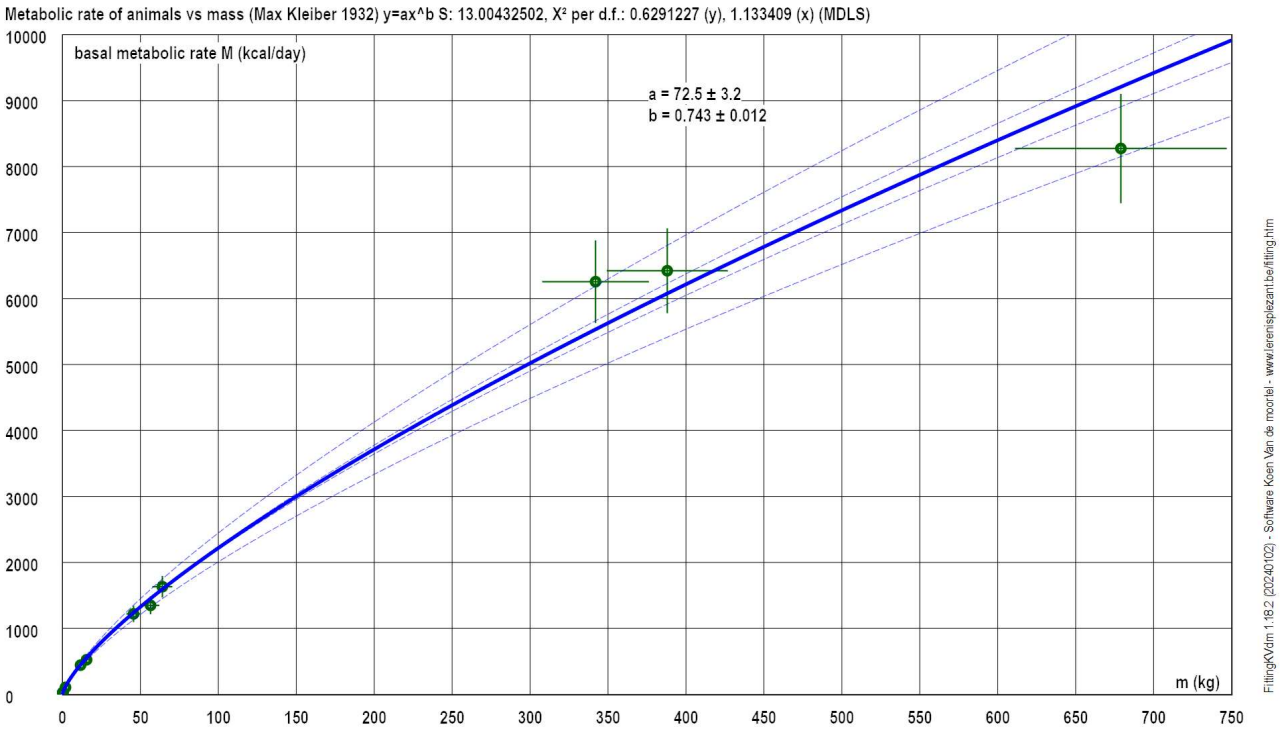


*** Animal_metabolism_vs_mass_Kleiber_1932.dta1**

Variables: x = body mass (m, in kg) of a mammal, y = basal metabolic rate (M, in kcal/day) of the animal. Errors of 10% were used (no real standard deviations available).

Source: measurements by the Swiss biologist Max Kleiber.
https://hilgardia.ucanr.edu/fileaccess.cfm?article=152052&p=VOWQRB

Bigger mammals consume more energy, but how much?
Try the "Power" model to find the relationship! With MDLS you come close to the theoretically expected relationship $M \sim m^{0.75}$.

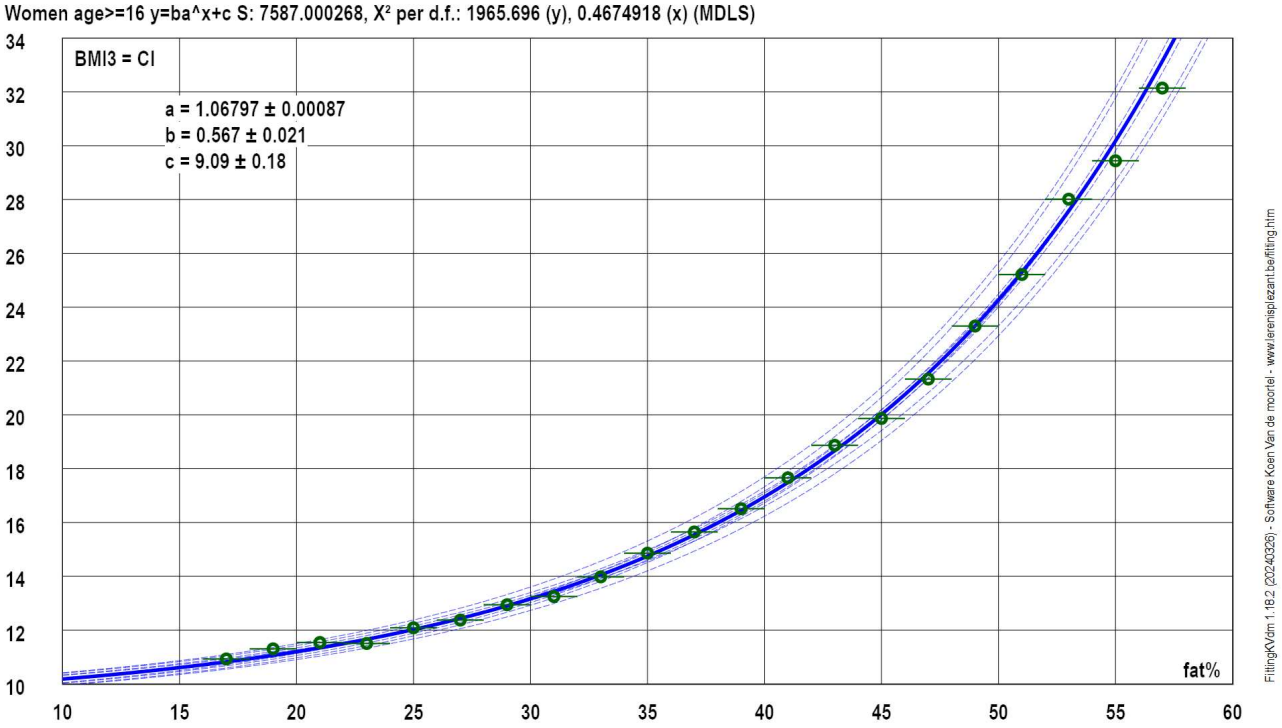


* **BMI2 vs fat Men 16 and older DIRECT DATA.dta1, BMI2 vs fat Men 16 and older.dta1**
* **BMI2 vs fat Women 16 and older DIRECT DATA.dta1, BMI2 vs fat Women 16 and older.dta1**
* **BMI3 vs fat Men 16 and older DIRECT DATA.dta1, BMI3 vs fat Men 16 and older.dta1**
* **BMI3 vs fat Women 16 and older DIRECT DATA.dta11, BMI3 vs fat Women 16 and older.dta1**

Variables: x = body fat percentage; y = the classical BMI (Body mass index = "BMI2" = m/h^2 with m: body mass in kg, and h: height in m) and "BMI3" (Corpulence index = m/h^3). Files with "direct data" contain all the individual data, while in the others they are grouped in bins of fat percentages [16, 18[, [18, 20[etc.

Source: 8039 American men, age 16 and more, 7475 women, NHANES

There seems to be a more or less exponential relationship between the corpulence index and the body fat%. For example:



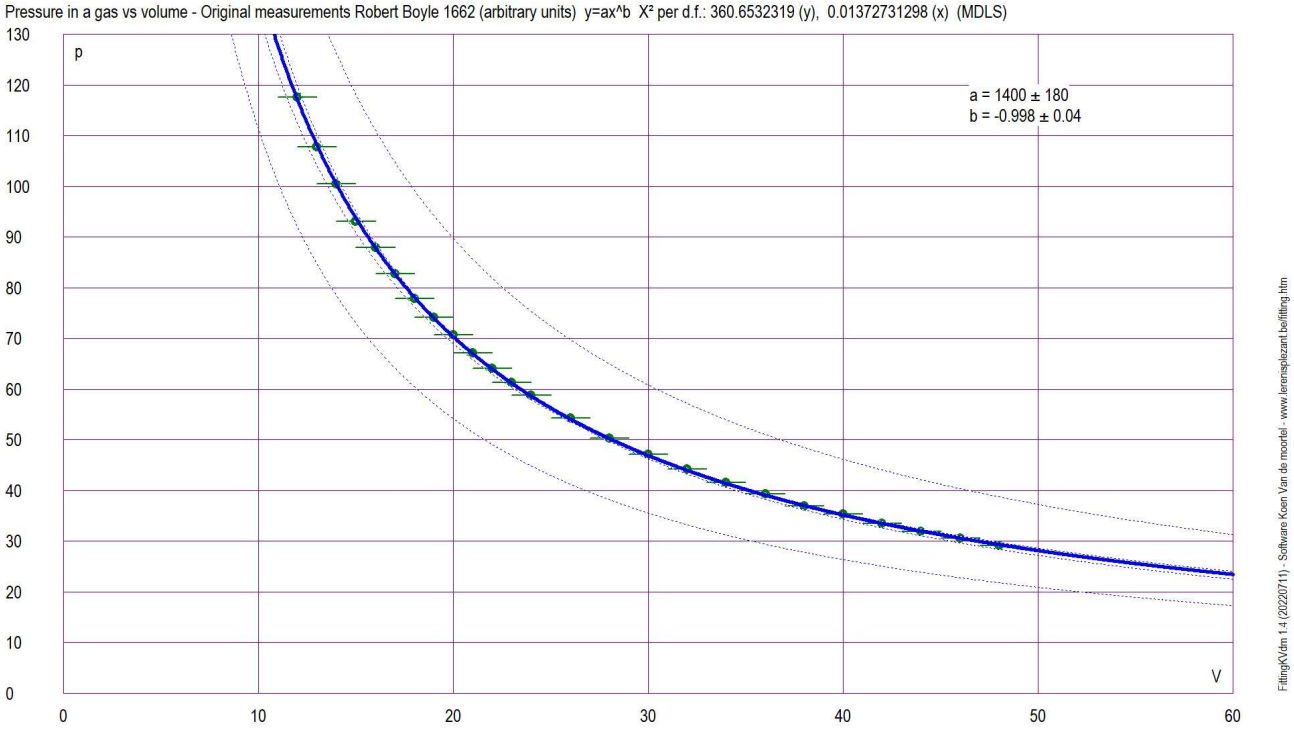
See also:
www.researchgate.net/publication/358736303_The_Body_Mass_Index_recalculated

* Boyle.dta1

Variables: x = volume (V), y = pressure (p) in a gas, arbitrary units. No information about the precision was given, so I assumed the pressure was ±1 and the volume ±0.0125 since all measurements were multiples of this value.

Source: Original measurements Robert Boyle 1662, cited in:
web.lemoyne.edu/~giunta/classicalcs/boyleverify.html

Of course, the appropriate model here is "Power".
The purpose of the measurements was to prove that the pressure is inversely proportional to the volume, or in other words: the exponent b should be -1.
With MDLS you get -0.998 ± 0.040, with OLS: -0.996 ± 0.055.

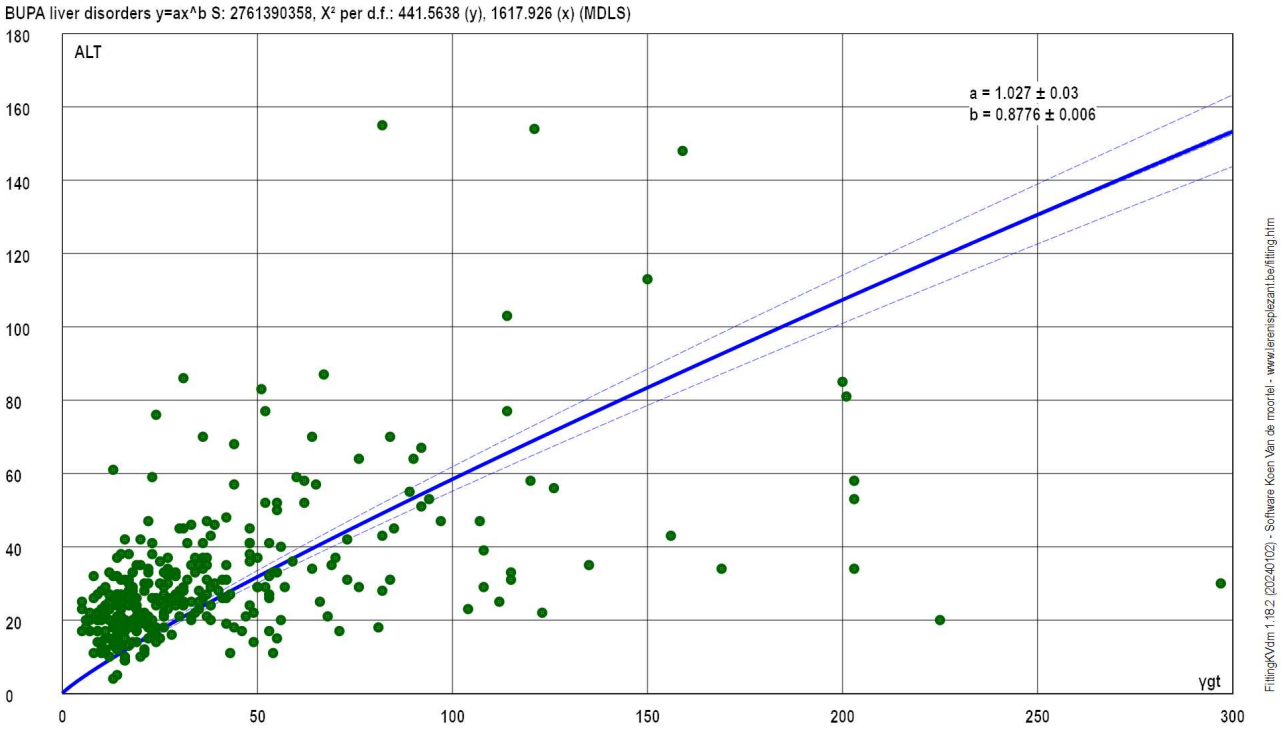


* **BUPA ALT vs GGT.dta1**
* **BUPA GGT vs ALT.dta1**

Alanine aminotransferase (ALT) and glutamyltransferase (GGT) are markers in the blood that might predict liver disorders, see: www.ncbi.nlm.nih.gov/pmc/articles/PMC3633107/. They are supposed to be related by a power function. You can check that for yourself. Some data, measured by "The British United Provident Association Limited" (BUPA), were collected from this site: web.archive.org/web/20171023174701/http://ftp.ics.uci.edu:80/pub/machine-learning-databases/liver-disorders/

Variables: x = ALT, y = GGT and vice versa in the second set.
Strangely no units were mentioned in the dataset, but I assume it must be concentrations, some quantity per liter. All the numbers were rounded to integers; I assumed errors of +-1 by lack of given confidence intervals.

This is how the power function fit looks. The correlation is very poor (Kendall τ is only 0.40). Also try OLS in both directions to see the dramatic differences.

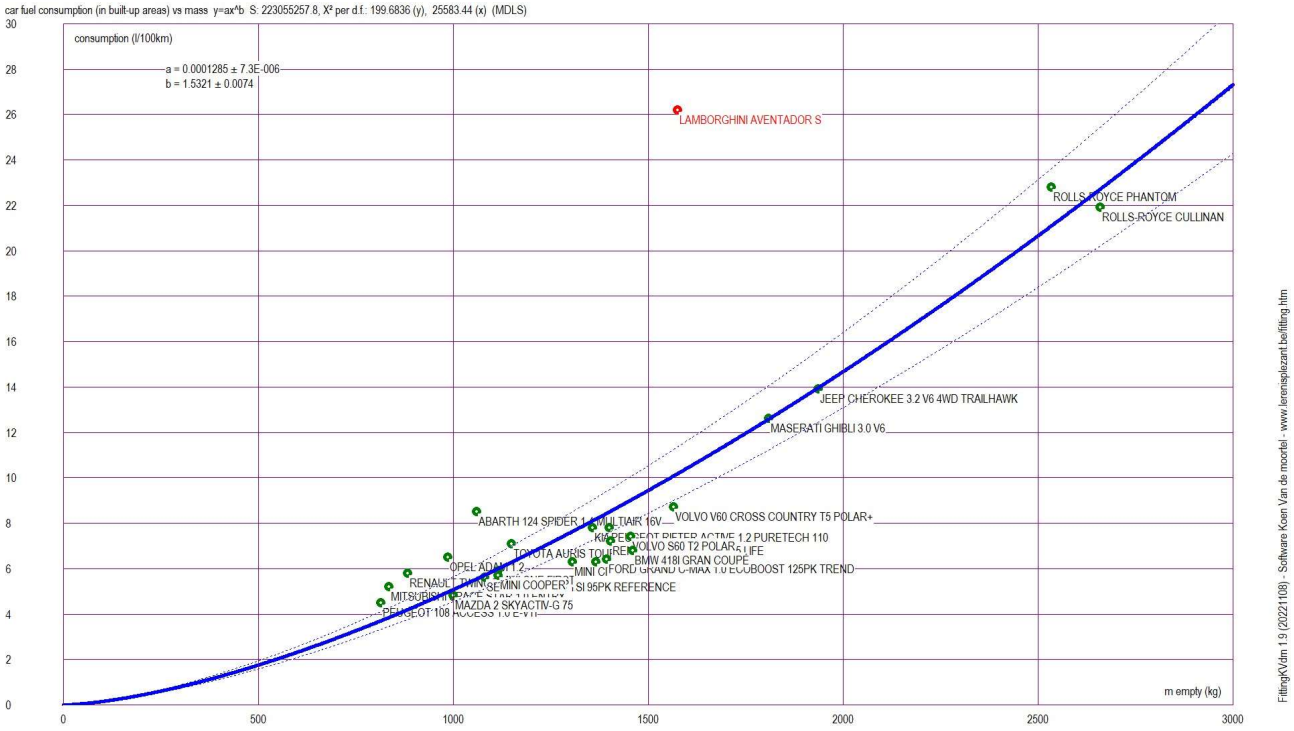


* Car fuel consumption vs mass.dta1

Variables: x = empty mass of a car (in kg); y = average benzine consumption (in l/100km) of the car when driving in urban areas.

Source: https://www.autoweek.nl/carbase/?bouwjaar=2018--2021&brandstof=benzine

A power function fits reasonably well, with exponent about 1.5.
One type of car seemed to consume way above expected: the Lamborghini Aventador S, so to make conclusions about more "normal" passenger car behavior, I left that one out for the fitting.



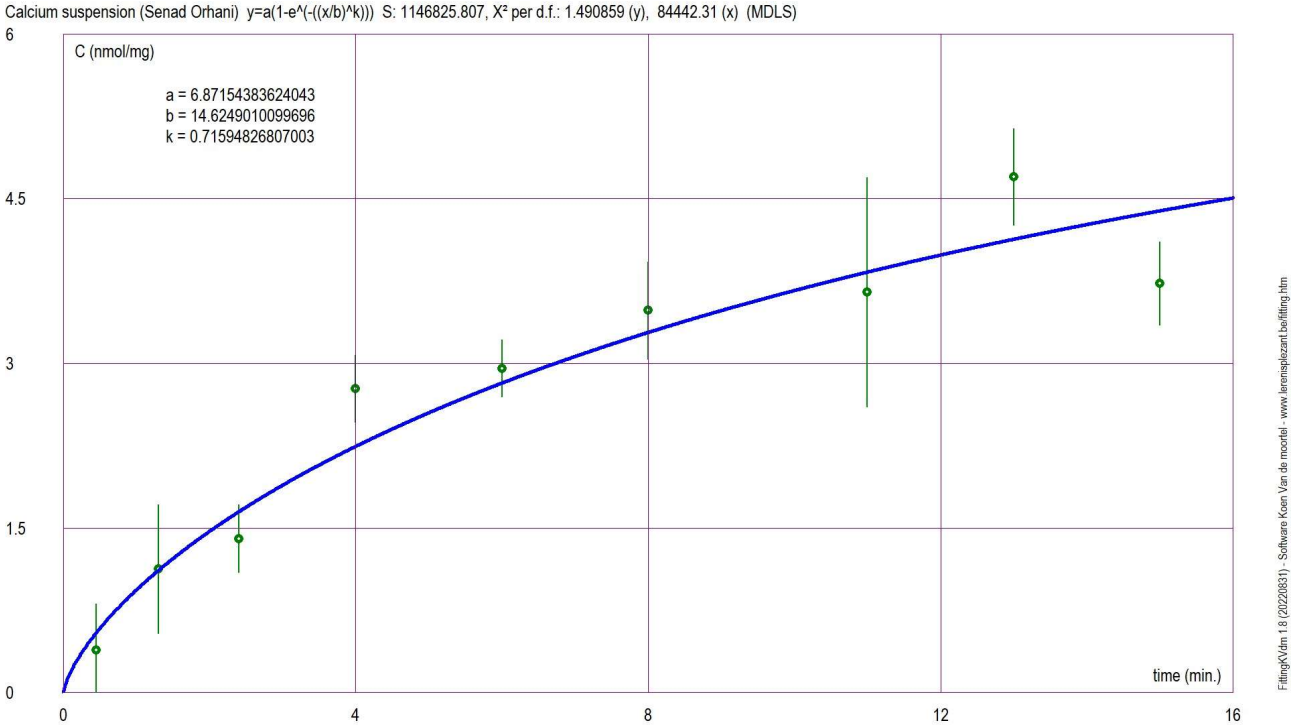
A quadratic function with c=0 also fits well, but I'm not sure if that makes more sense. It is less sensitive to the Lamborghini outlier though.
Some sources suggest an exponential relationship, but that would be weird, since a zero mass should have a zero consumption. I have even seen a guy on Youtube explaining a linear regression on the "consumption" expressed in the American way (miles per gallon) vs mass; that is crazy of course!

* Ca-suspension.dta1

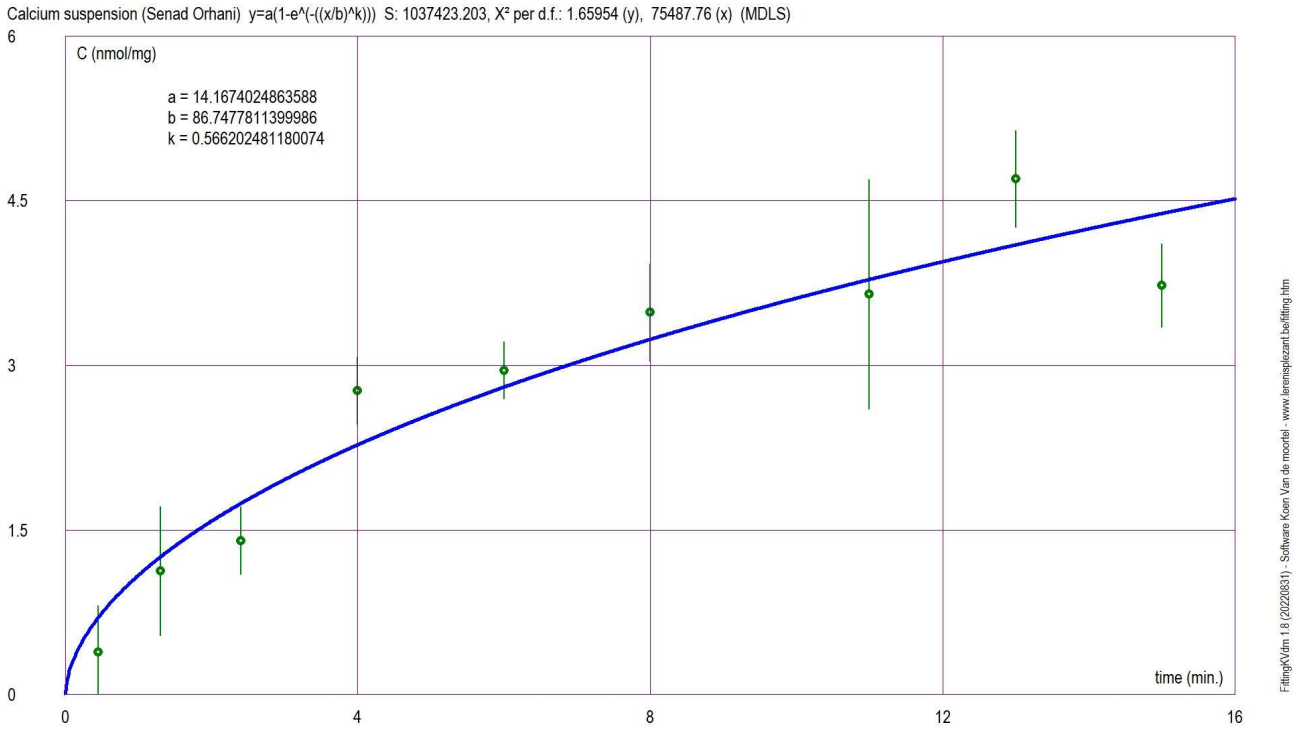
Variables: x = time in minutes, y = concentration of suspended calcium (nmol/mg).
No precision was given, but for each time value, 3 measurements were done, so I calculated the standard deviation to be used as the precision. For the time, I assumed equal precisions 0.01min.

Source: this was presented as an example for Weibull growth by Lazim Kamberi, Alejna Alimi and Senad Orhani: "Nonlinear models by parameters and their transformation" (doi: 10.55059/ijm.2022.1.3/49), see:
www.researchgate.net/publication/361983006_Nonlinear_Models_by_Parameters_and_their_Transformation

This is a good example to show that the "Weibull growth" is not a stable function to fit! You need a serious amount of precise data to make a good estimation of the parameters. To understand this, compare the first graph, obtained with MDLS after 25 iterations:

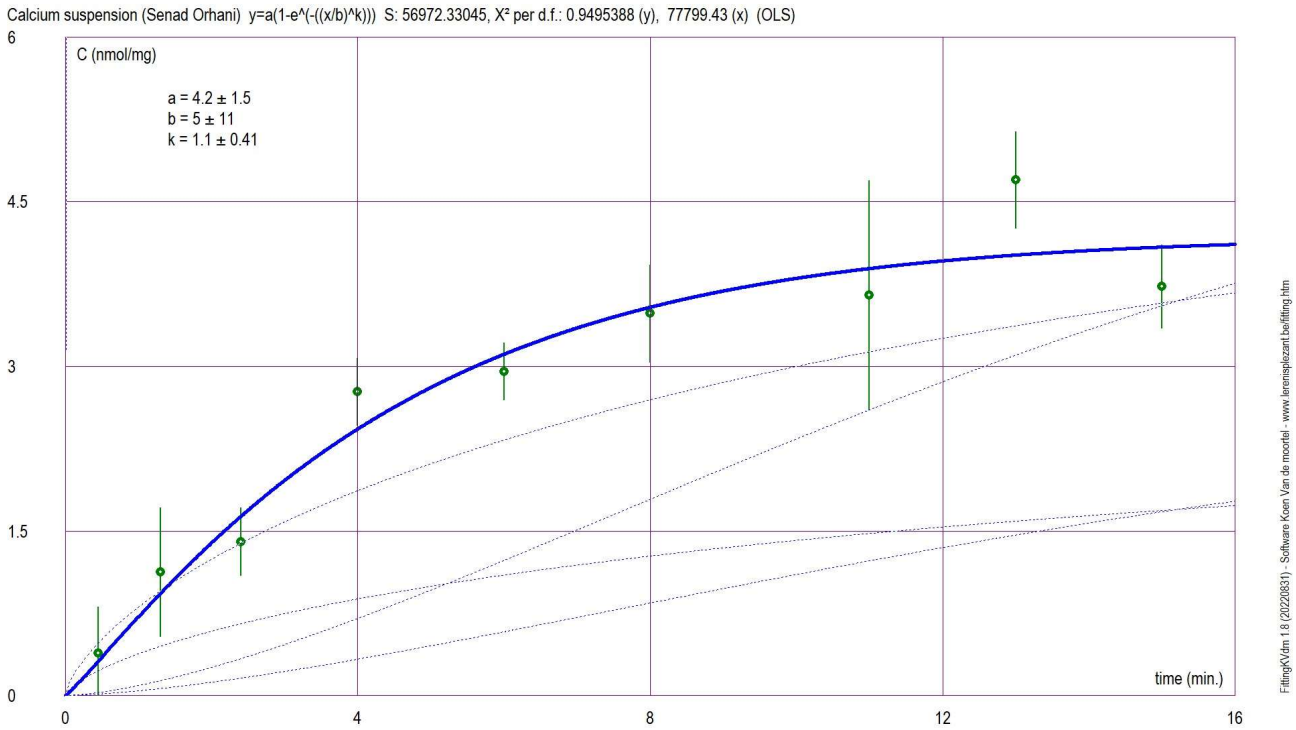


and the second one, after 200 iterations:

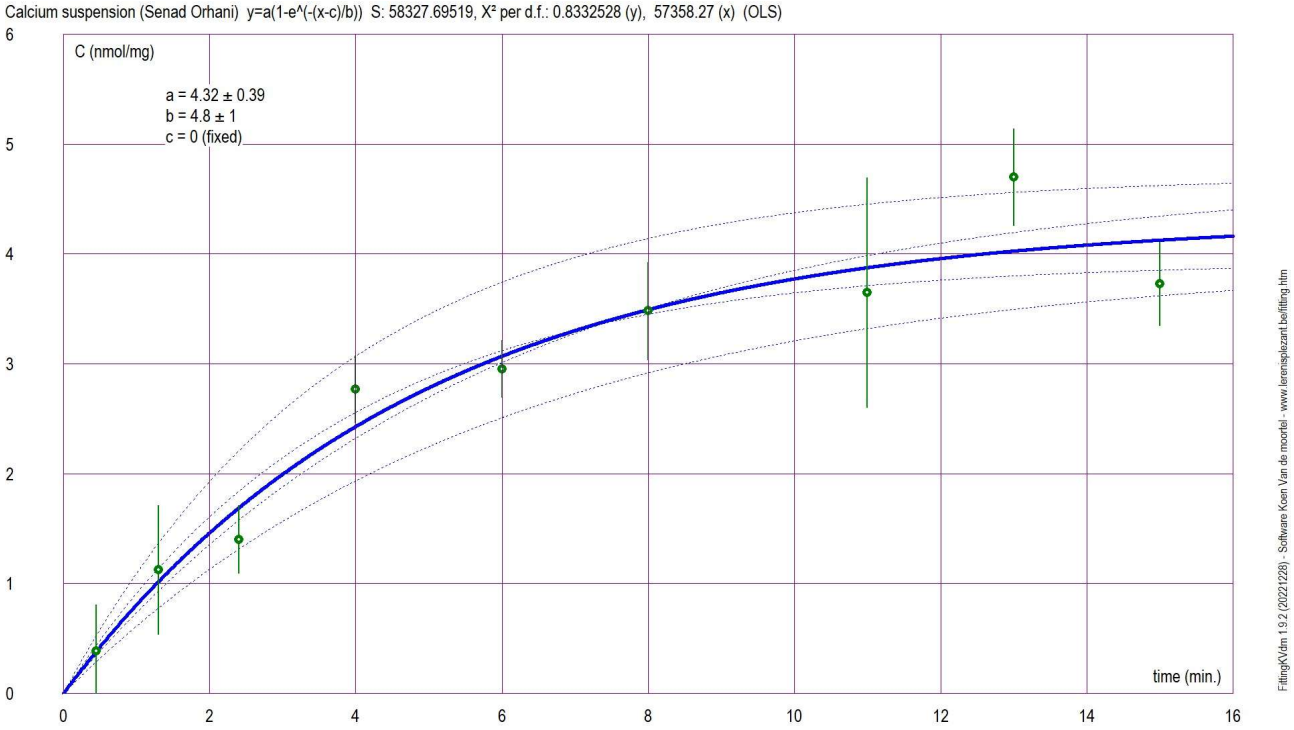


The curves are almost the same, though the parameter values are very different. And after more iterations, they will diverge even more. The right conclusion is that this model is not reliable in this case, especially when parameter k is free floating! Fixing k to a fixed value might be a good idea, maybe k=1 since there is no reason to believe there should be an inflection point.

With OLS, there is a better convergence, but that doesn't mean the parameters are more reliable. You can see that from the dotted lines, "worst case scenarios", based upon the data imprecisions:



It might be wiser to just use a simpler model here, like “Plateau - exponential”, with c fixed to 0, since the concentration starts from 0.



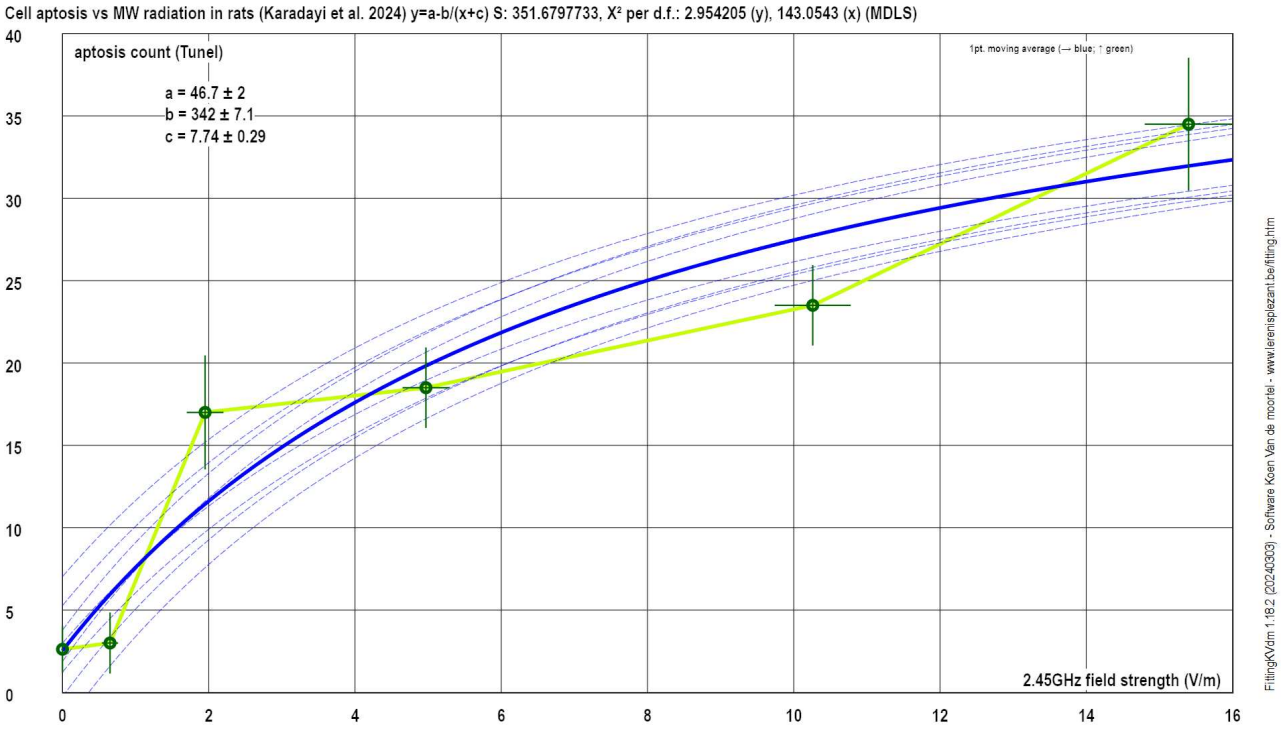
If the last values are in the horizontal asymptote area, as it seems to be, it is safer to use OLS. This model predicts a final concentration of 4.32, not very different from the Weibull model, but this one is 3 times more precise!
MDLS is also stable here, but it predicts a final value slightly above the maximal concentration. More measurements would be better, to be more sure about this asymptotical value.

*** Cell aptosis vs microwave field strength with rats (Karadayi 2024).dta1**

Variables: x = field strenght (V/m) of 2.45GHz radiation that a number of rats were exposed to (1h/day during 45 days); y = number of dead cells in slices of bone tissue, using the so called "Terminal deoxynucleotidyl transferase dUTP nick end labeling" method (short: TUNEL) that detects DNA fragmentation from aptosis (programmed cell death).

Source: Karadayi, A. et al.: *Does Microwave Exposure at Different Doses in the Pre/Postnatal Period Affect Growing Rat Bone Development?*, in: Physiol Res. 2024 Mar 11; 73(1):157-172. See: www.biomed.cas.cz/physiolres/pdf/73/73_157.pdf

From the small number of measurements, it's not sure which model fits best, but a homographic one ("Rational 2") makes sense:



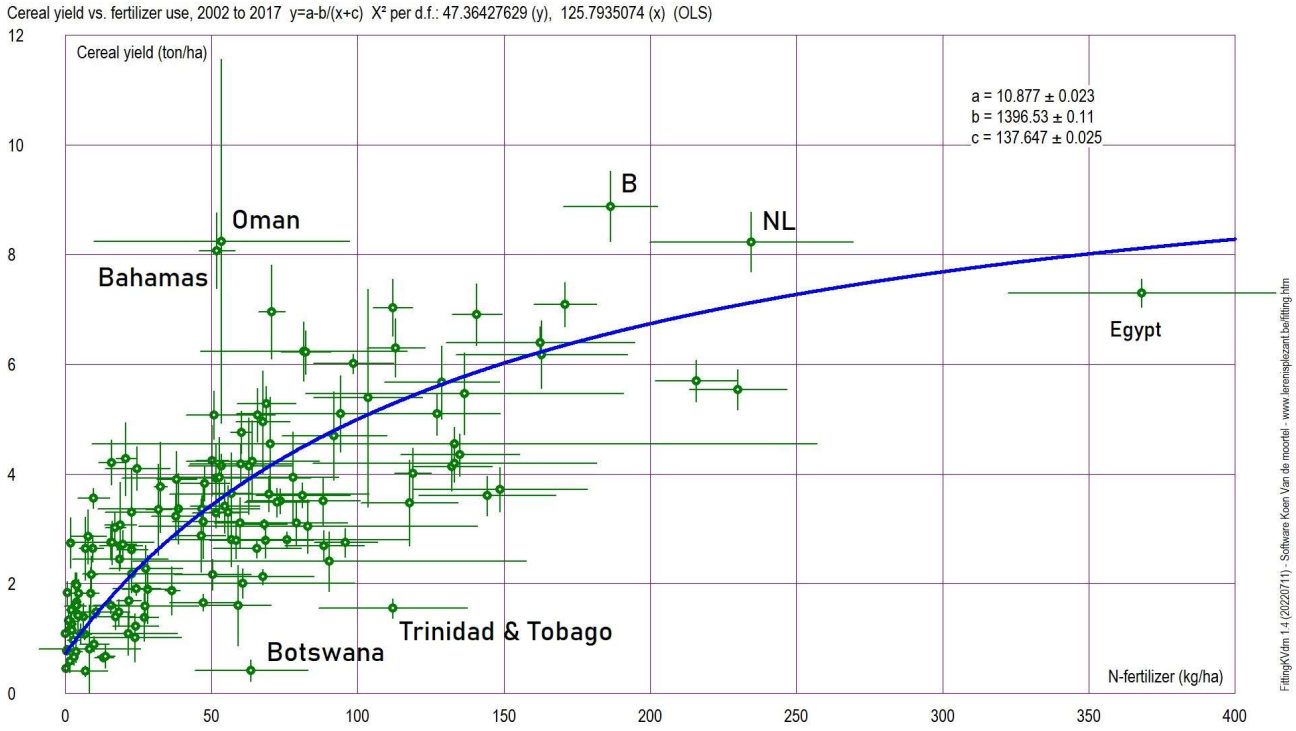
Obviously, even such small doses seem to have an adverse health effect.

* Cereal-crop-yield-vs-fertilizer-per-country.dta1

Variables: x = applied amount of nitrogen fertilizer (kg/ha), y = cereal yield (tonnes/ha). For each country, the averages and standard deviations of both variables (error flags!) were calculated from 2002 to 2017.

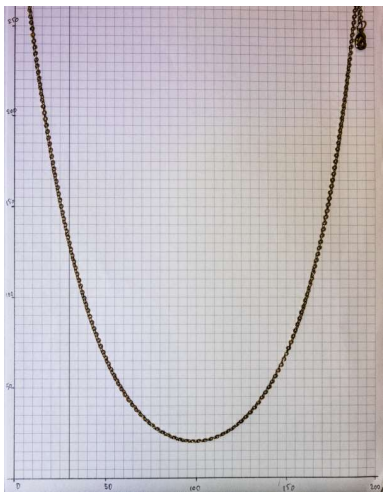
Source: https://ourworldindata.org/grapher/cereal-crop-yield-vs-fertilizer-application

If a limit in the yield is expected, models like "Exponential+baseline" or "Rational 2" can be used. Since some countries seem to produce near this limit, OLS is preferred here.
The "Rational 2" model predicts a maximal yield of 10.877 tonnes/ha. Feel free to try other models.



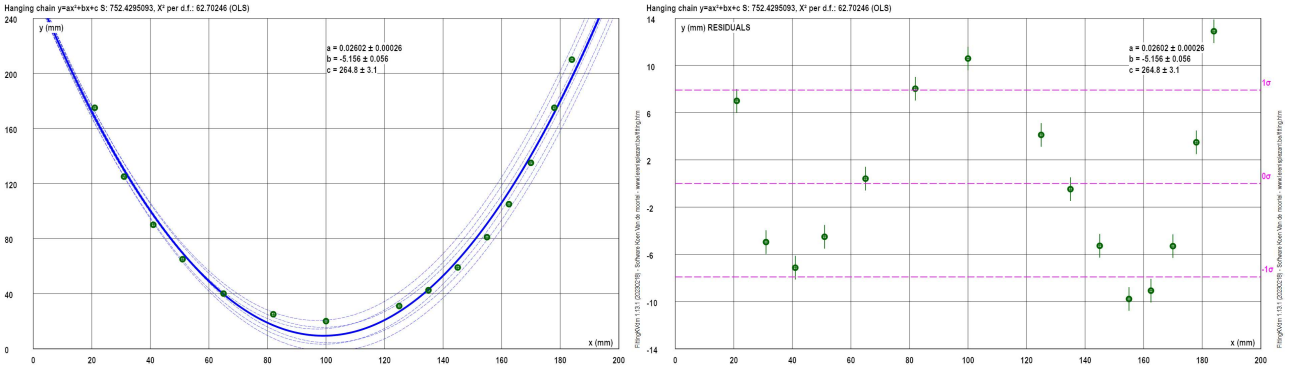
Note: the pattern is not very clear, since obviously many other variables play a role here, especially the climate, quality of the seeds, diseases, the soil, etc. etc. That's why the fitting needs a lot of iterations!
Conclusions that can be made here: points above the curve represent countries that are doing well (Belgium, Netherlands, but especially Oman and the Bahamas since they need much less fertilizer). Egypt has good results but uses way too much fertilizer which is very bad for the soil and the nutricious quality of the crops. The lowest points are countries that perform very badly, e.g. Botswana.

* Chain.dta1

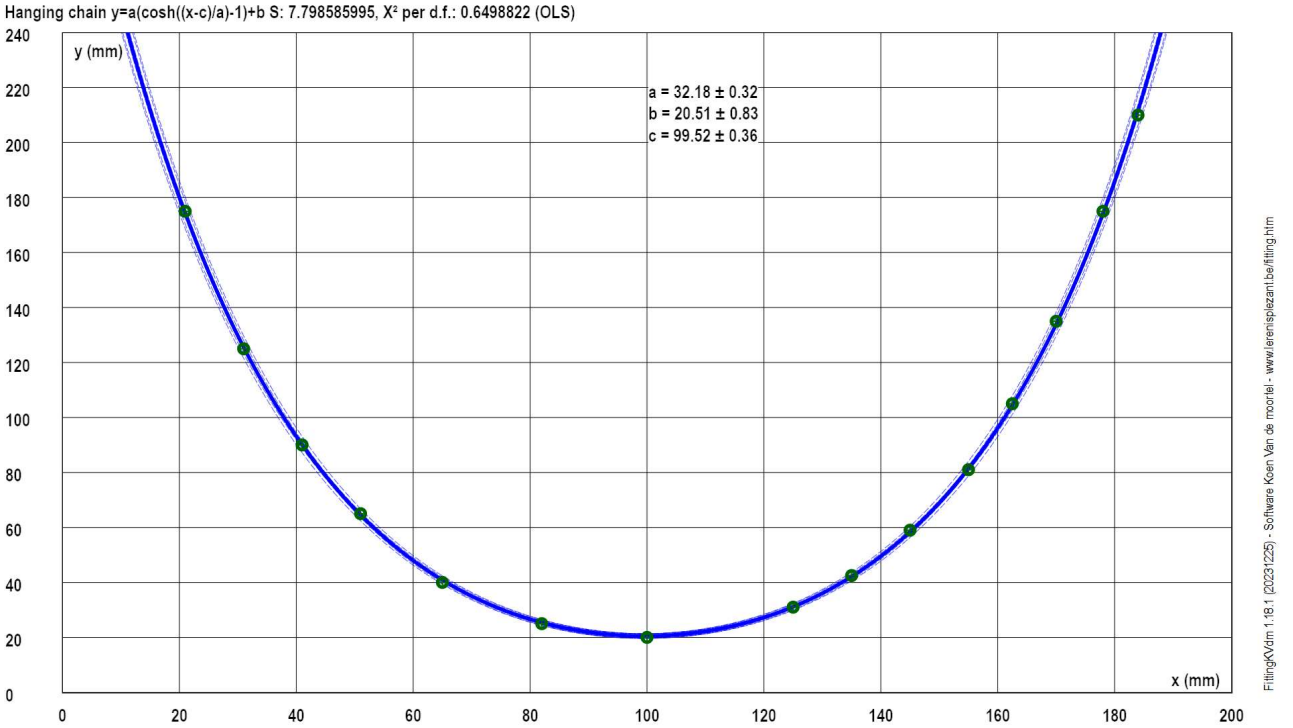


Variables: x and y are the coordinates of points of a hanging chain (necklace), measured on a photograph (see right). Some contrast enhancement and lens correction was done to ease the measurements.

At first sight, you might think the curve looks like a parabola, but fitting with a “Quadratic” model shows that it must be something different.
This is also clear by looking at the residuals: there is a clear (non-random) pattern visible!



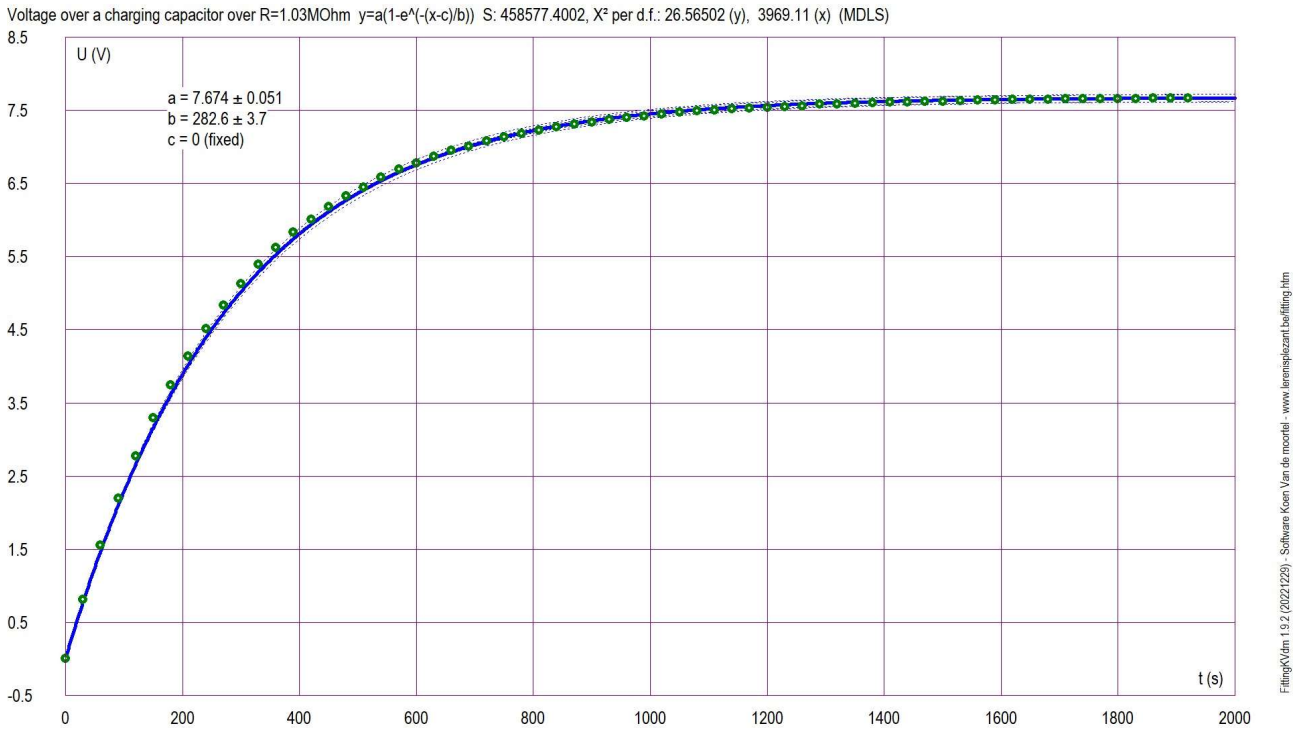
The only well fitting model here, is - of course - the “Chain line” model.



* Charging_capacitor.dta1

Variables: x = time (t, in seconds); y = voltage over a capacitor that is charged over a resistor R = 1.03MΩ.

Try the "Plateau - exponential" model to find the source voltage and the capacity! It will confirm quite well what the theory predicts.



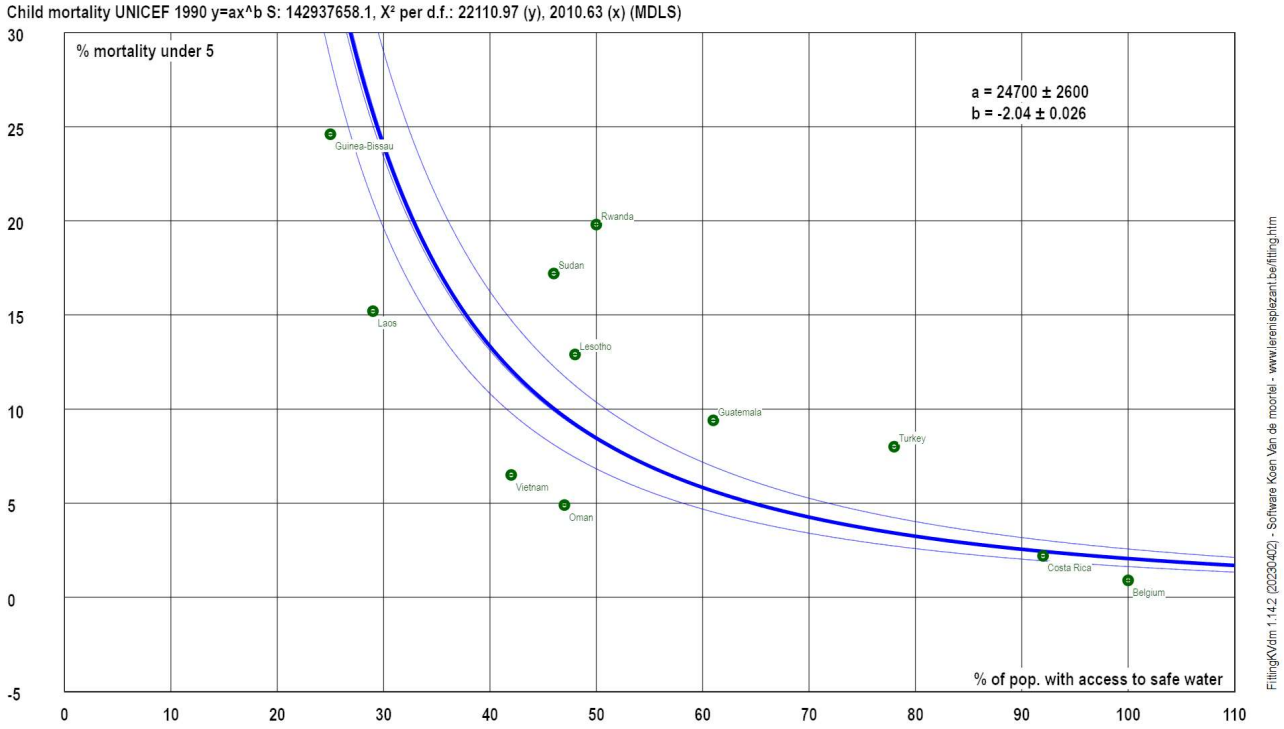
Since b = RC, we can conclude that the capacitor had a capacitance of 282.6s/1.03MΩ ≈ 274µF.

* Child mortality UNICEF 1990.dta1

Variables: x = the percentage of the population with access to safe water (1988-90); y = The under 5 mortality rate (percentage of annual deaths of children under five years of age, measured in 1990 or the latest available year before then). The measurement imprecisions were assumed to be half of the least significant digit.

Source: "A Handbook of Small Data Sets", by D.J. Hand, F. Daly, A.D. Lunn, K.J. McConway and E. Ostrowski, Springer 1994, quoting a UNICEF research. I added one missing x value for Belgium because I'm pretty sure that everybody had running water at home there in 1990, even my grandparents!

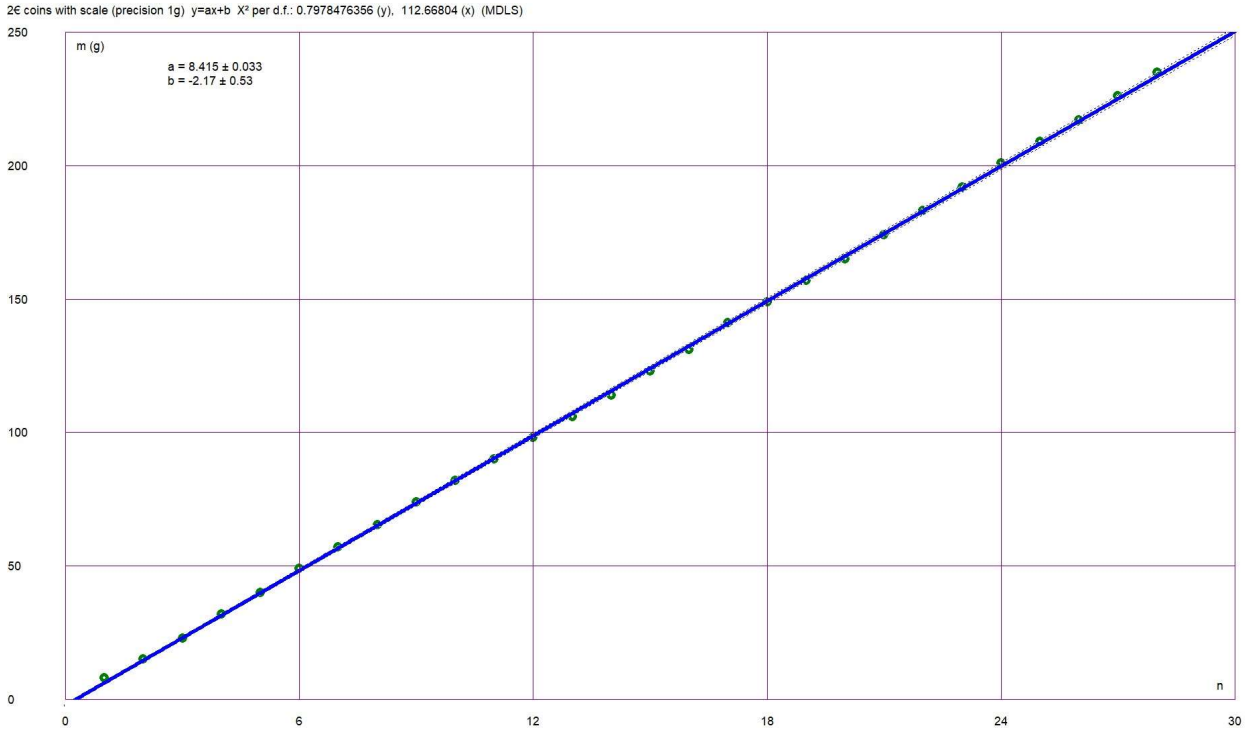
Appropriate model? Many number crunchers just fit a straight line through the dots without thinking, but that makes totally no sense if you extrapolate it.
Probably no model is perfect here, since x and y are both percentages, but at least a power function with a negative exponent makes more sense, if you neglect the parts where x>100% or y>100%.
Try and compare MDLS and OLS! MDLS seems way more realistic, especially for high x values!



* Coin mass.dta1

Variables: x = number (n) of 2€ coins; y = their total mass (m, in g).

Of course, the linear model is most appropriate here, using MDSL, because there is a real (invertible) relationship between the number of coins and the total mass. Notice however, a small bias of the (cheap) scale: the line doesn't go through the origin precisely. So the slope of this line is a better estimation of the average coin mass than the one you get from just weighing 29 coins and dividing by 29.

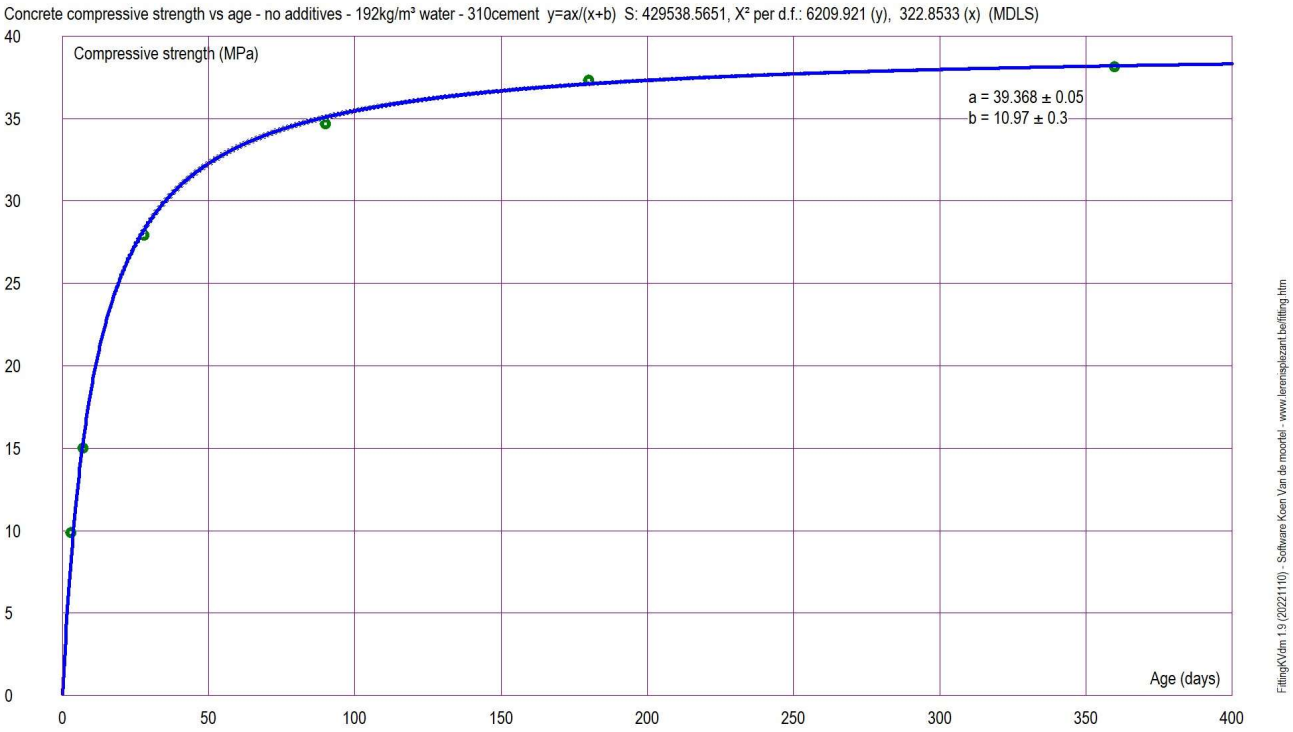


* Concrete compressive strength vs age - no additives - 192 water.dta1

Variables: x = time (in days) after a specific concrete mixture was made; y = compressive strength after x days of hardening (in MPa). The uncertainty on x was unknown, hence set to 1; the original y values must have been derived from something else (kilograms?) so the uncertainties were not sure (my guess: 0.01). Anyway they were probably all the same, so the absolute values of σ_x and σ_y only matter for the parameter uncertainty estimation.

Source:
data.world/uci/concrete-compressive-strength/workspace/file?filename=Concrete_Data.xls
The selected mixture had 310kg cement, 192kg water, 970kg coarse aggregate, 850kg fine aggregate, no additives.

A fitting model should obviously start from the origin, since concrete starts from a muddy mixture, and a horizontal asymptote: the maximal strength after a long time.
"Rational 1" seems to fit quite well.

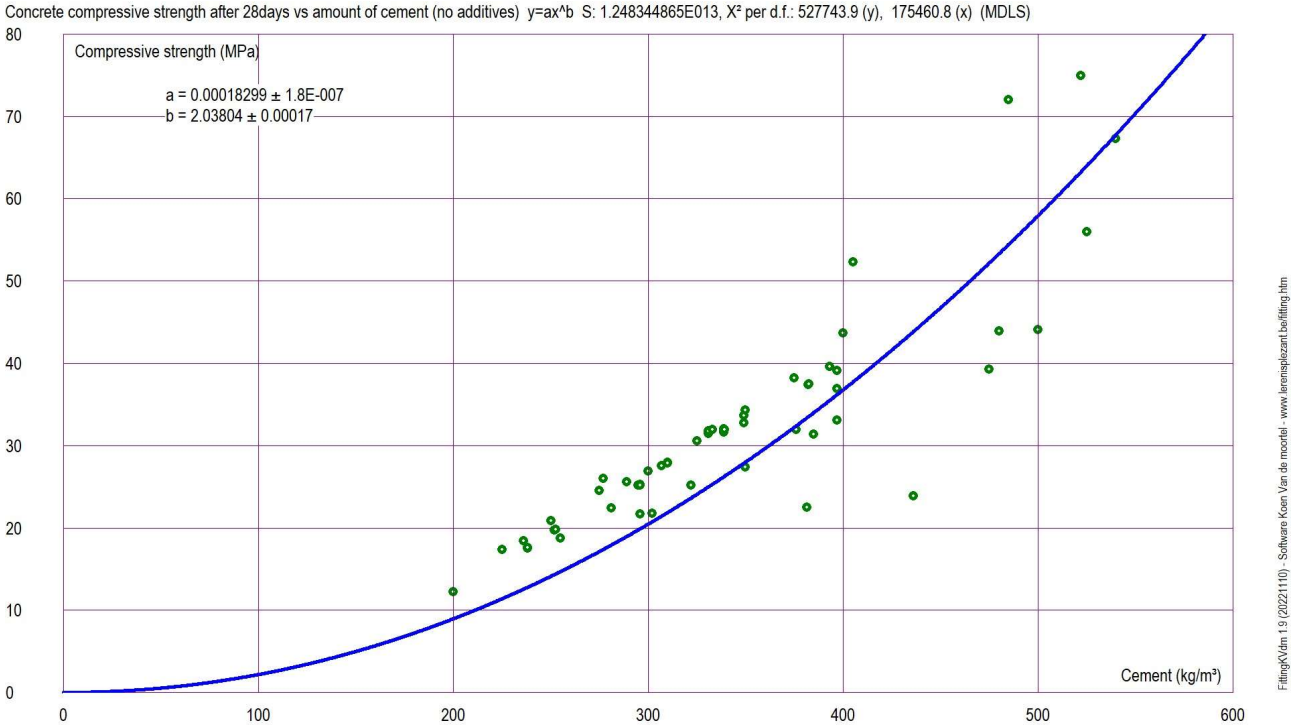


* Concrete compressive strength vs cement - no additives - age28.dta1

Variables: x = amount of cement (in kg) in a 1m³ concrete mixture; y = compressive strength after 28 days of hardening (in MPa) (see previous example). The uncertainty on x was set to 0.1 since many of the measurements were measured with that precision.

Source: see previous file. Mixtures without additives were selected; the amount of water varied between 146-203kg, coarse aggregate: 838-1125kg, and fine aggregate: 594-945kg.

A fitting model should obviously start from the origin, since no cement = no concrete.
The "Power" model suggests a more or less quadratic relationship (lower exponent if you use OLS), but it's clear that other variables play a role here. E.g. most of the points above the curve have less water than those below (click "Show labels" to see this).
The "diode" model produces almost the same curve.
It would be more interesting to have measurements from mixtures with more cement. What would happen then? A maximum??? Anyway the model is not much extrapolatable anyway because there is only a limited amount of cement that can go in 1m³.

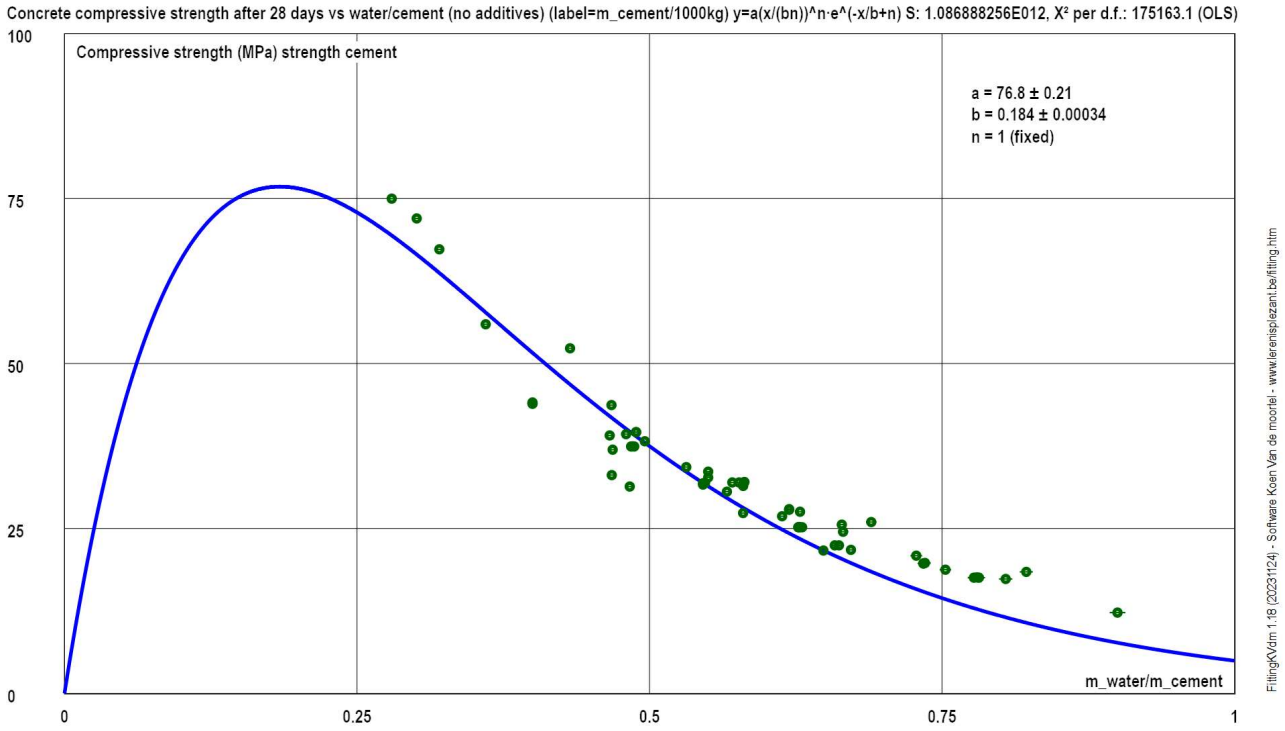


*** Concrete-compressive-strength vs water-cement ratio age28 no add.dta1**

Variables: x = ratio of the amount of water and the amount of cement (both in kg) used in a concrete mixture; y = compressive strength after 28 days of hardening (in MPa) (see previous example). The uncertainty on x was calculated assuming the uncertainty on the masses was 1.

Source: see previous file. Mixtures without additives were selected; the amount of water varied between 146-203kg, coarse aggregate: 838-1125kg, and fine aggregate: 594-945kg.

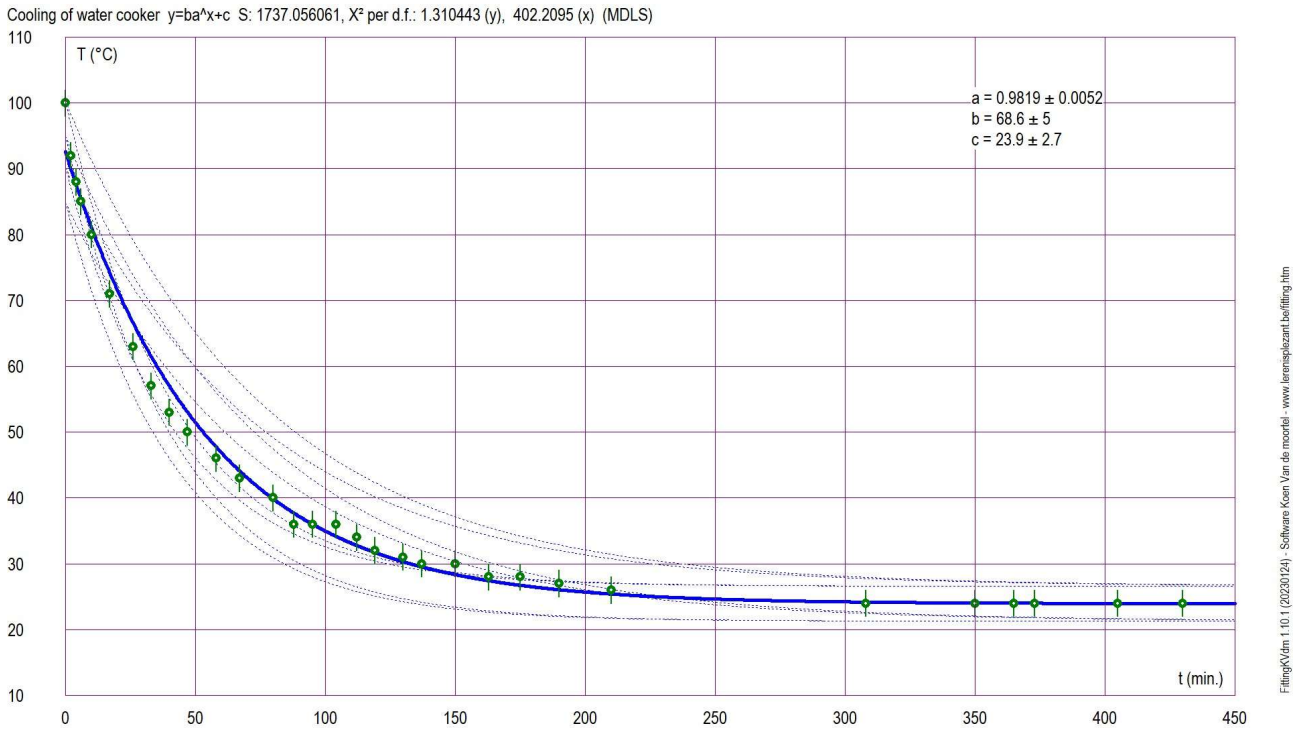
A fitting model should obviously start from the origin, have an optimal value and then go back to zero, since no water or too much water = no concrete. A model that has this behavior, is "Power•Exp.decay". Since there are no data from the actual peak, it's best to fix parameter n to a simple value like 1.



* Cooling_of_water_cooker.dta1

Variables: x = time (t, in minutes); y = temperature (T, in °C) of the water in a cooker, measured with a bad thermometer in the cooker itself.

Try the "Exponential + baseline" model! And MDLS since there is a real relationship between the time and the temperature (Newton's cooling law).

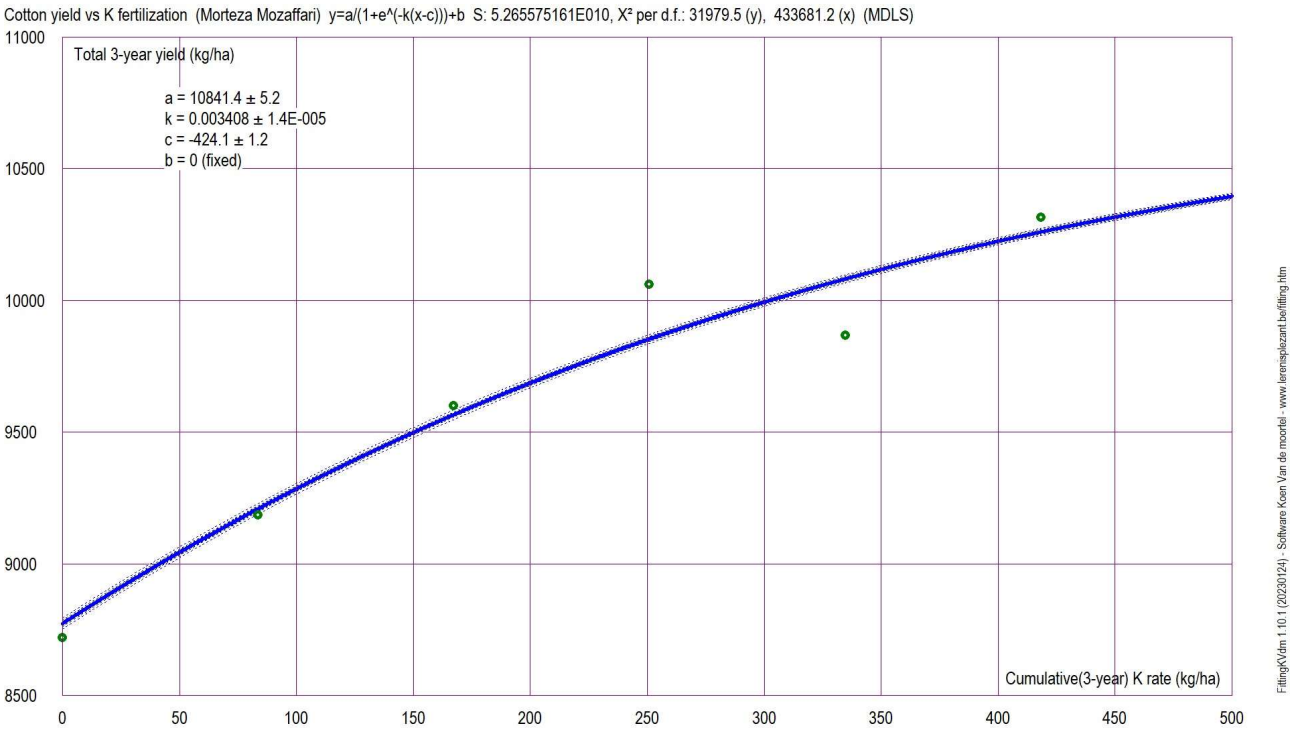


* Cotton yield vs K conc Morteza Mozaffari.dta1

Variables: x = cumulative (3-year) fertilization with potassium (K) rate (in kg/ha); y = total 3-year yield (kg/ha) of cotton on this field. The $\sigma_{x,i}$ and $\sigma_{y,i}$ values were estimated here only from the measurement precision, not taking into account the yearly variability, so the estimation of the precision the parameters will be too good.

Source: Morteza Mozaffari: www.scirp.org/journal/paperinformation.aspx?paperid=85771

Many models will give a reasonable fit, but only those with a horizontal asymptote ("Logistic + baseline" with b=0, "Exponential + baseline", "Plateau minus exponential", "Rational 2", "Transition") make sense, since the effect of more fertilization disappears once the plants have enough. The logistic and transition models are the most logical, since on the left side it goes to negative values of fertilization. My guess is that that part of the graph might give an indication of the amount of potassium in the soil without fertilization; that's for the agricultural engineers to investigate. The limit value L = 10841kg/ha is the maximum possible yield according to the logistic model.
The convergence is very slow here, because the points are too much aligned. It would have been better to do at least one more measurement with more fertilizer.

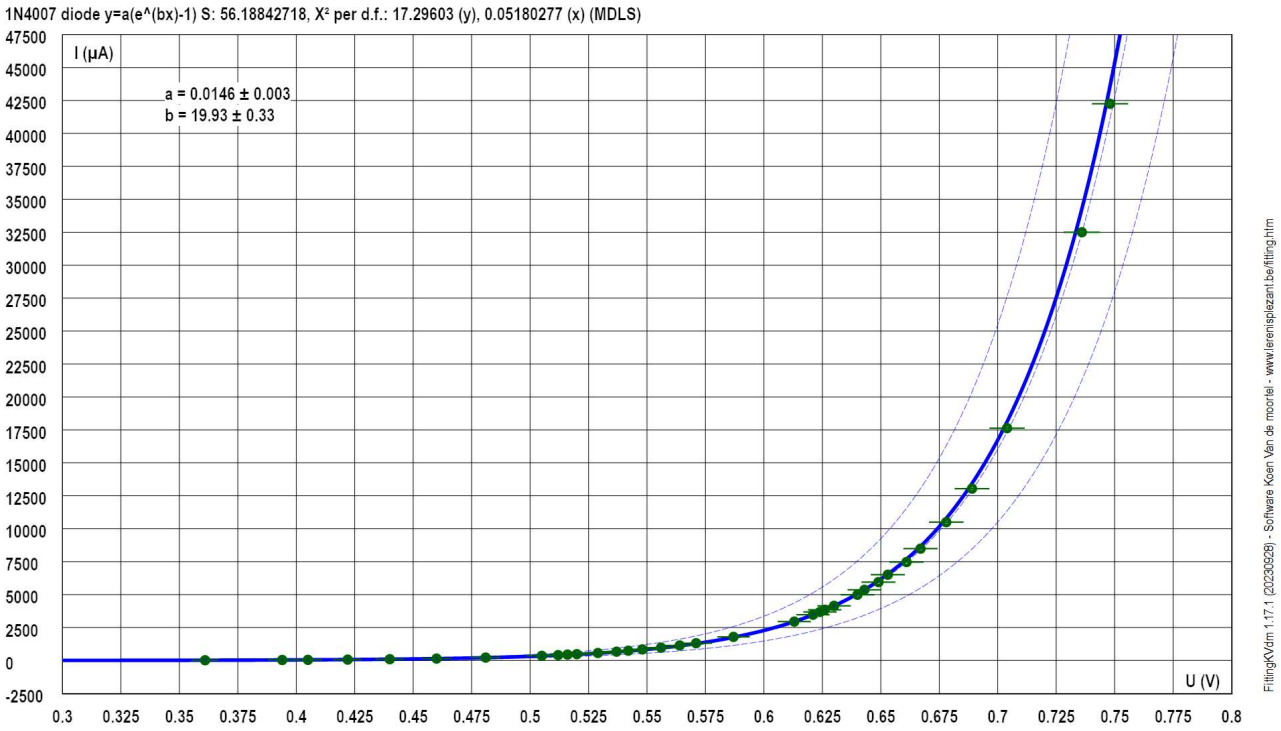


Remark: Here you can find similar data from all countries:
https://ourworldindata.org/grapher/cereal-crop-yield-vs-fertilizer-application
But, these data are not useful to determine the right model, since all the possible influences are mixed together here: each country has a different soil, climate, cereal species, etc.; each year has different weather etc. Also: these measurements do not include the very important reference: the yield without fertilization.

* **Diode Si 1N4007.dta1**

Variables: x = voltage (in V) over a silicon diode (type 1N4007), y = current through it (in µA).
The voltage was measured with an Ohmeron MT488B in the range 0..4.000V. The accuracy according to the manual should be 0.5%+4 counts = 0.5%+0.004V.
The current was measured with an Owon OW18E multimeter.
Accuracy in the 200.00µA range: 0.5%+10 counts = 0.5%+0.1µA.
In the 2.0000mA range: 0.5%+1µA; 20.000mA range: 0.5%+10µA; 200.00mA range: 0.5%+100µA. The "10 counts" are certainly not exaggerated, since the displayed values went up and down quite a lot.

The appropriate model here is "Diode".
This predicts a maximal reverse current of 0.0146µA, which is better than the datasheet gives as "worst case" at 25°C: 5µA when 1000V is applied. Direct measurement with a reverse voltage of 12V gave a value of less than 0.01µA (unmeasurable).
See: https://www.diodes.com/assets/Datasheets/ds28002.pdf
Remark: if you do the measurements with reversed voltage, you will see some current in the order of 1µA, but that is due to the internal resistance of the multimeter (usually about 10MΩ), so don't do that.
Since the measurements were relatively precise, OLS and MDLS don't differ much.

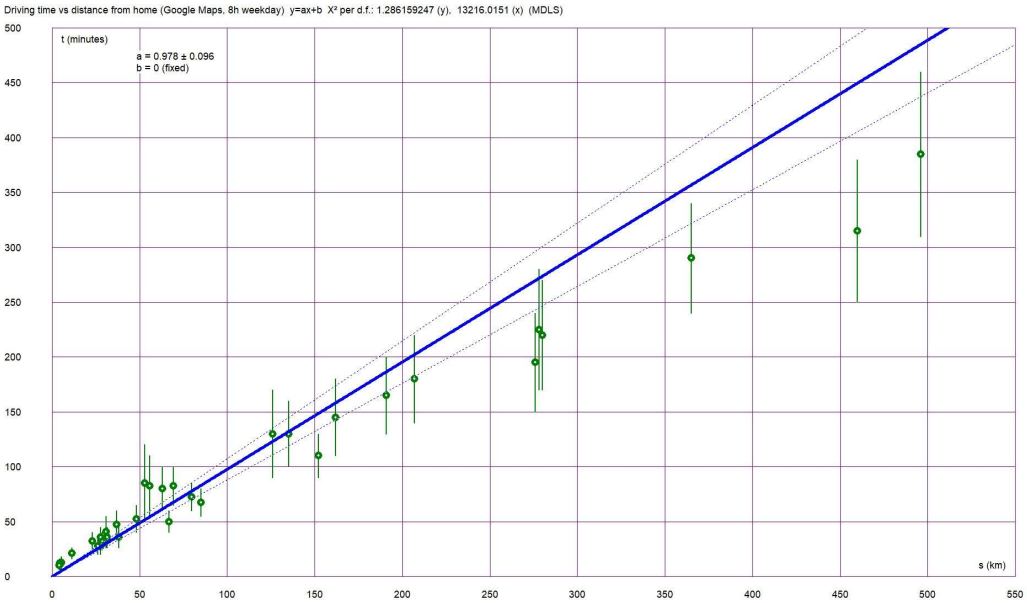


The same measurements were done with a germanium diode and a white LED, but they don't fit in the "ideal diode" model at all! You can try with these data files:
* **Diode Ge OA72.dta1**
* **Diode White LED.dta1**

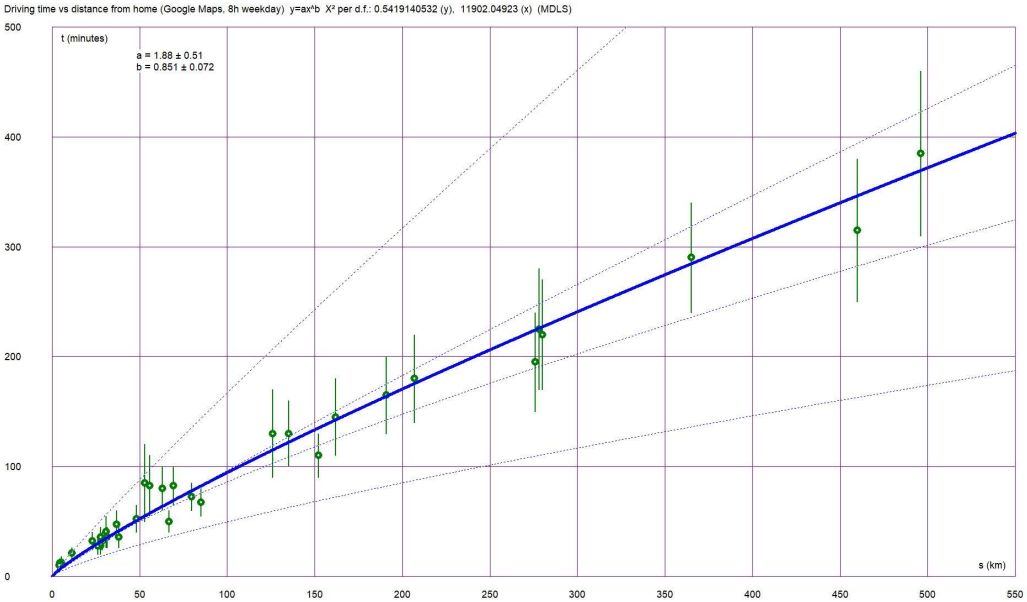
A possible explanation might be that the temperature at the pn junction raises significantly, even with a few mA of current, so the parameter b changes. This is difficult to measure, but keeping the diode temperature as constant as possible (using cooling equipment) might possibly help.

* Driving times vs distance.dta1

Variables: x = distance from my home, by road, in km; y = average driving time according to Google Maps (start 8am on a working day, σ_y = half of the difference between the longest and shortest time).
You can try the "Linear" model, but don't forget to fix b = 0 since zero distance is always zero time. If you doubt to use MDLS or OLS, ask yourself if it is ok to ask the inverse question "How far can I drive in a given time span?". Of course that's ok, so use MDLS!



But, you will see that the "Power" model (with b < 1) fits better! Why? For longer distances you will usually drive more on highways, which increases the average speed and hence decreases the driving time.

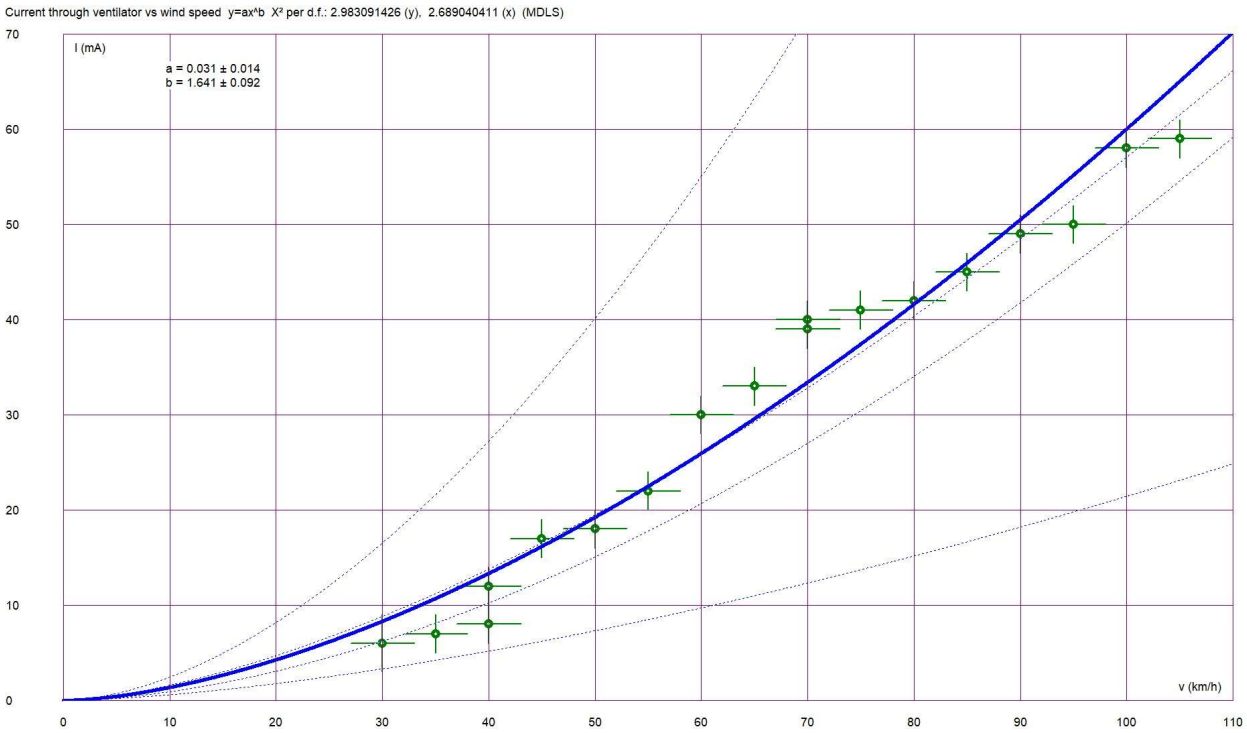


Try also OLS if you want to see the dramatic difference!

* Electrical current from wind.dta1

The experiment: On a windless day, a simple small computer ventilator was kept outside the window of a moving car, perpendicular to the driving direction to catch the wind maximally. It was attached to an Ampere meter. (Not so easy to do this precisely, I can assure you!)
Variables: x = the speed of the car (v, in km/h), y = current (I, in mA).

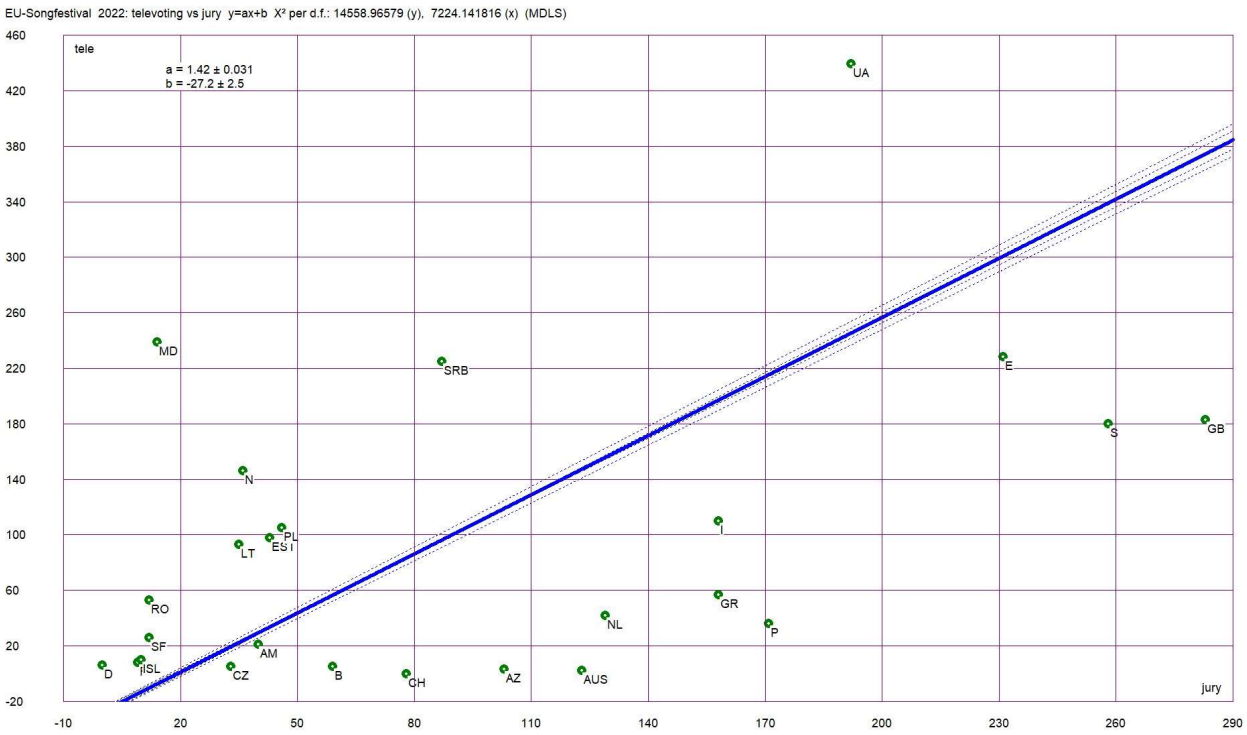
Theoretically, the generated power (P) should be ~ v^3, and $P=RI^2$ (R = internal resistance of the meter), so $I \sim v^{1.5}$. But also the characteristics of the ventilator (not designed to be used as a generator) play an unknown role of course.
See: https://en.wikipedia.org/wiki/Wind_power



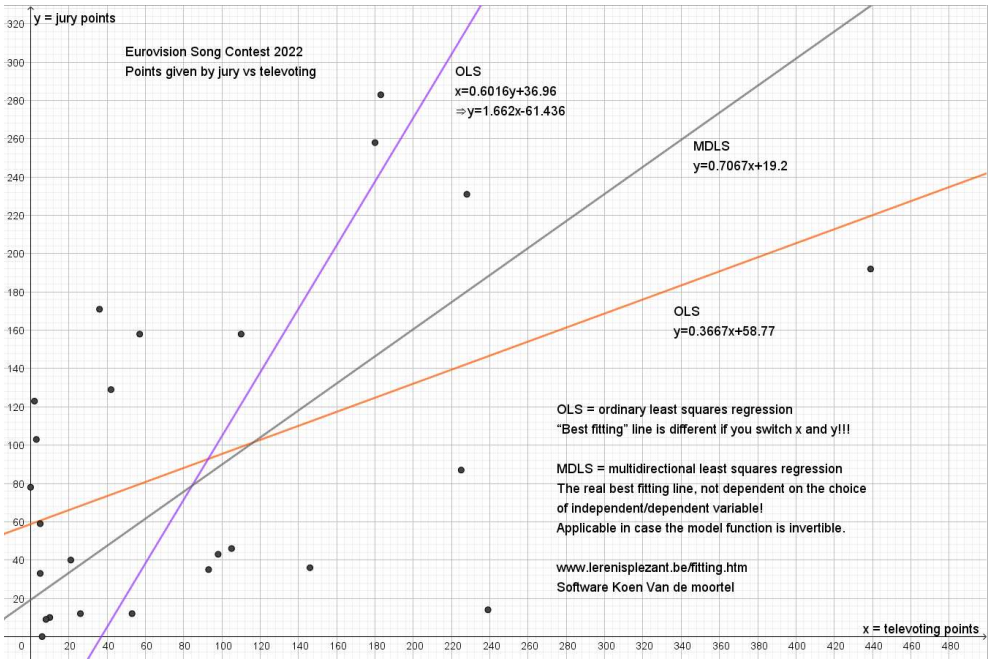
* **Eurovision Song Contest 2022 jury vs tele.dta1**
* **Eurovision Song Contest 2022 tele vs jury.dta1**

Variables: x = points given by televoting, y = points given by the jury at the Eurovision contest of 2022. Vice versa in the second file.
Source: https://eurovisionworld.com/eurovision/2022

If the points were given completely honestly, one would expect more or less y ≈ x, but in reality the correlation was weak: the dots in the graph are very scattered. It is possible to find a linear estimation of the relationship between x and y, but definitely use MDLS! If you use OLS, switching x and y will produce ridiculous differences, see: https://www.researchgate.net/publication/360620480_Another_example_why_multidirectional_regression_is_better_than_OLS_the_2022_Eurovision_Song_Contest_points



The graph on the right summarizes it:

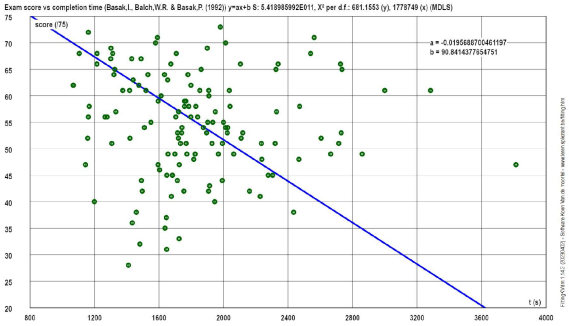


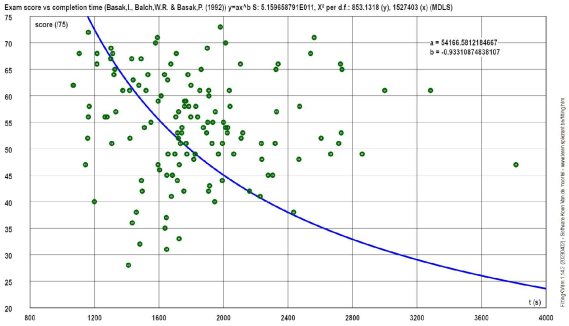
* Examination scores vs completion time.dta1

Variables: x = time to complete an exam (in seconds); y = score (max = 75).

Source: “A Handbook of Small Data Sets”, by D.J. Hand, F. Daly, A.D. Lunn, K.J. McConway and E. Ostrowski, Springer 1994, quoting an example from I. Basak et al. (1992).

This is a typical example of a cloud of points in which you can recognize nothing or whatever you want (a line, a power function,...). MDLS and OLS give very different results since there is no clear pattern.
Kendall tau = -0.069 (click “Show” in the “Report” section), meaning that there is a very small downward trend in the data, but forget about making any other conclusion.



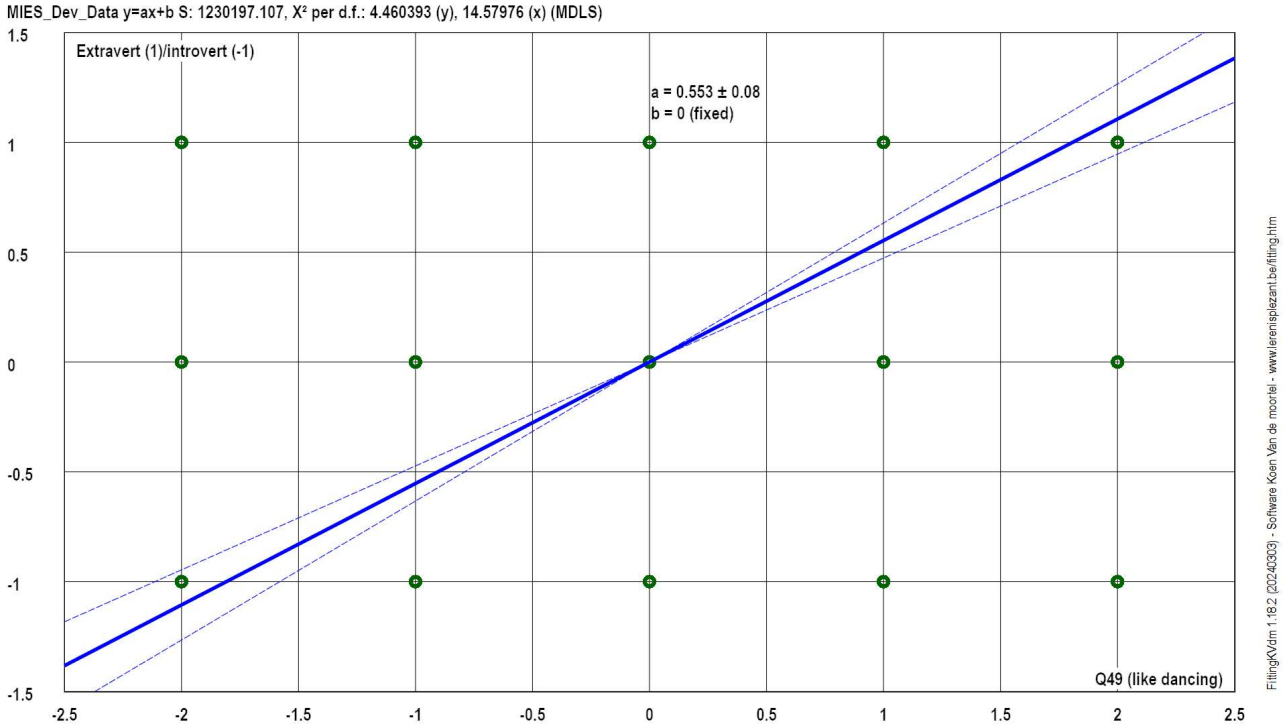


* **Extrav-Q4.dta1**
* **Extrav-Q49.dta1**
* **Extraversion Q4-Q49.dta1**

These data come from a questionnaire by Yam Peleg, that was made to find out if the answers to certain questions was related to a self-attributed extraversion rating: www.kaggle.com/datasets/yamqwe/introversionextraversion-scales.

The questions I selected Q4: "I would hate living with room mates.", and Q49: "I really like dancing.". In the first two files, x = answer to these questions ("disagree"… "agree", converted to -2..2) and y = extraversion rating (-1 if the person perceived himself as introvert, 1 if extravert, 0 if he didn't know).
In the third file, Q4 and Q49 are compared.

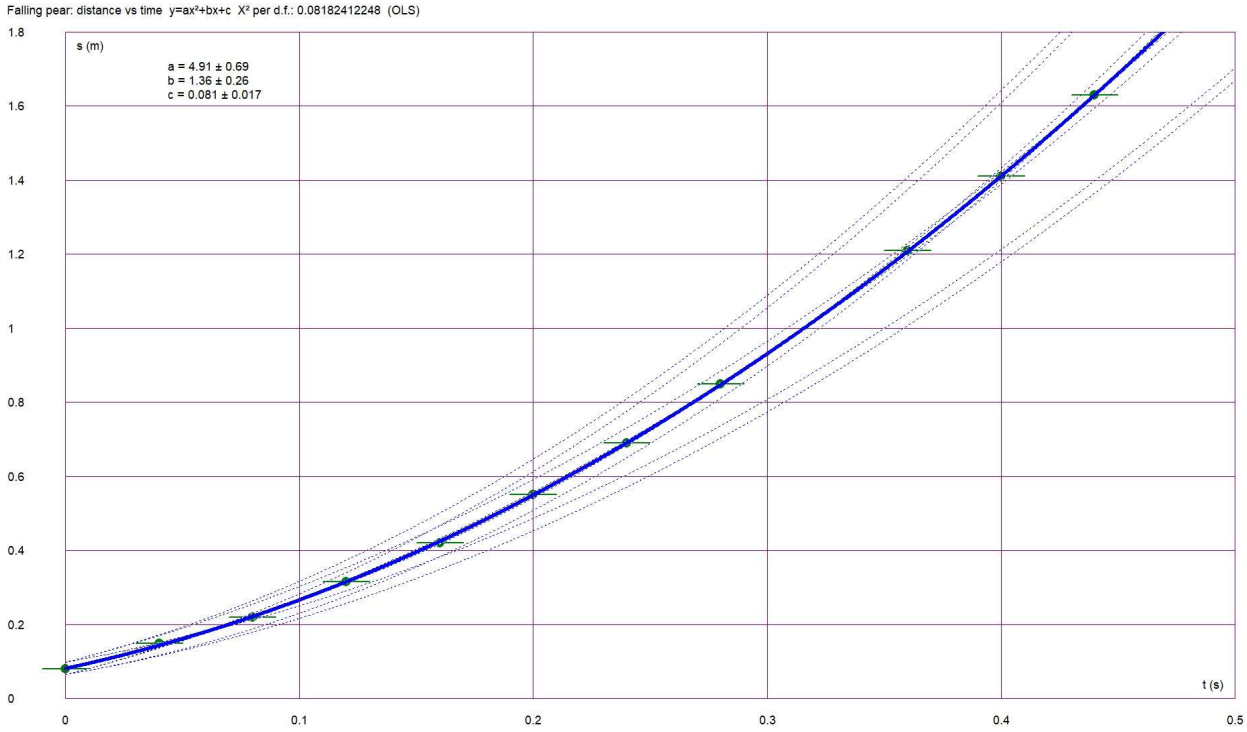
To avoid overfitting, only the most basic models can be used, like a linear one through the origin. All relationships are quite weak with very small Kendall tau and Pearson r values. For example:



* Falling pear.dta1

Variables: x = time (t, in seconds); y = vertical distance a pear fell down (s, in meters).
The measurements were done from a long exposure photograph while using a stroboscopic flash. Newton did it with an apple, but it works fine with a pear too!

Try the "Quadratic" model to find the value of the gravitational acceleration g (=2a)!
The meaning of c is here: the time the pear was already falling when the first flash was fired, and b was the velocity at that moment.



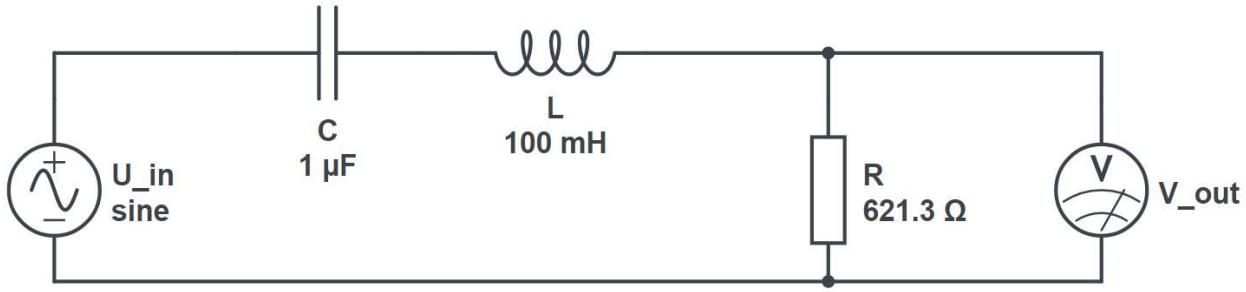
We can now predict that afters 10 seconds, the pear would have fallen 503m and have a speed of about 99m/s.

Predicted y value for x= 10 y = 503.187518571956

Derivative for x= 10 dy/dx = 99.2583760404583

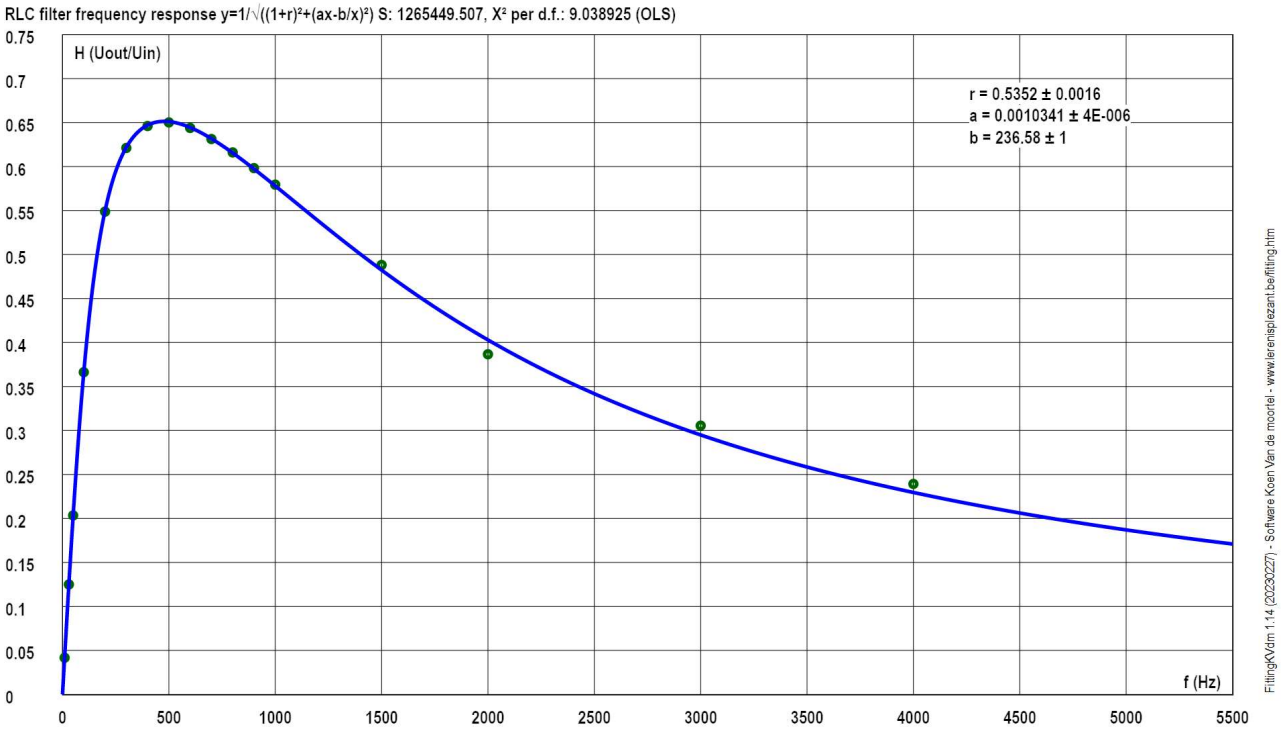
* **Filter 1 RLC.dta1**

Variables: x = frequency of a (more or less) sinusoidal signal; y = fraction of the voltage that passed through a serial RLC filter, i.e. U_{out}/U_{in}, with U_{out} measured over the resistor. The signal was taken from the headphone exit of a laptop pc, and generated with the Windows beep function (e.g. in Delphi Pascal: "beep(500, 10000);" generates a signal with a frequency of 500Hz during 10000ms = 10s). The voltages were measured with a simple multimeter (OWON OW18E).



The values of L and C shown on the schematic, were given by the manufacturer; R was measured.

The appropriate model for this bandpass filter is "RLC serial filter".



From the fitting we can derive:

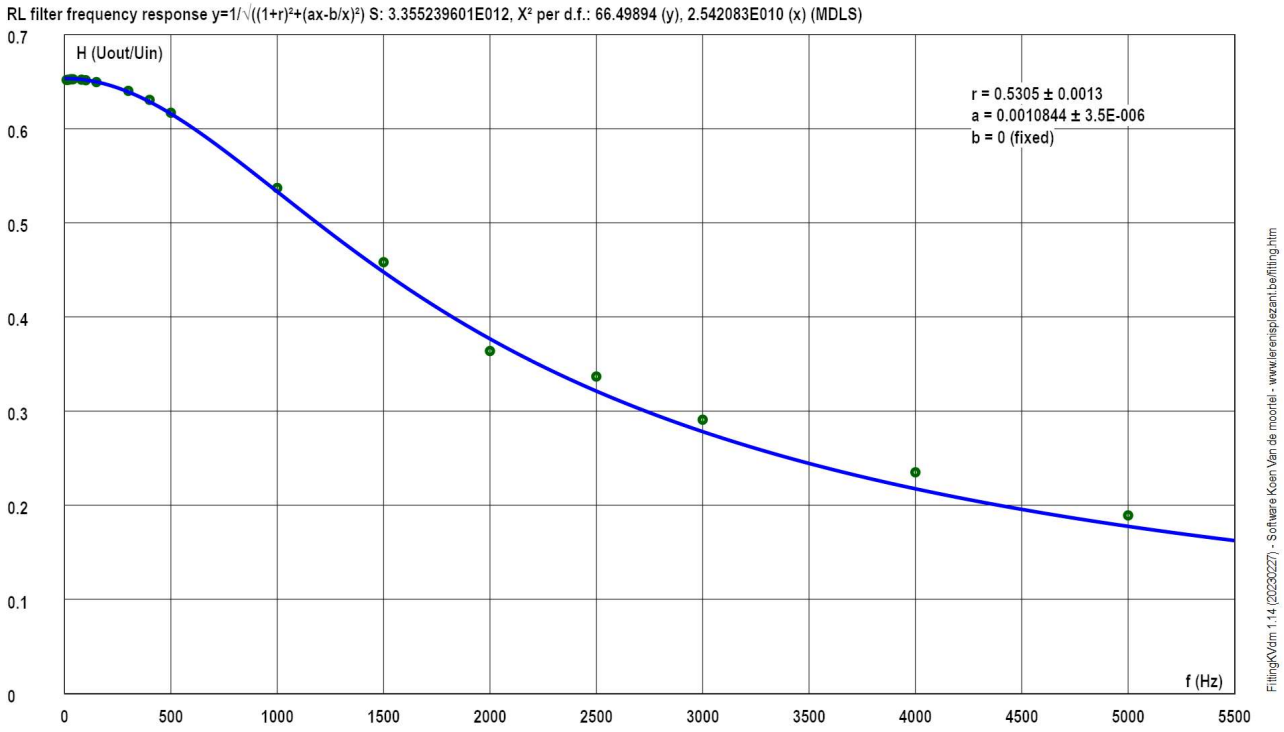
* The internal resistance of the coil: r·R = 0.5352 · 621.3 = 332.5Ω (measured with the multimeter: 332.6Ω);
* the value of C = 1/(2πR·b) = 1.08278µF (measured with multimeter: 1.0908µF, according to manufacturer: 1.0µF);

* the value of $L = a \cdot R/(2\pi) = 102.25\text{mH}$ (according to manufacturer: 100mH; not possible to measure with most simple multimeters.)

* Filter 2 RL.dta1

The setup was the same as the previous one, but without the capacitor.

We can use the same model, and the fit will normally converge to a very small b value. But it's better to set b = 0, or easier: select the model "RLC serial filter - preset RL". Now you can use MDLS, since there is no peak.

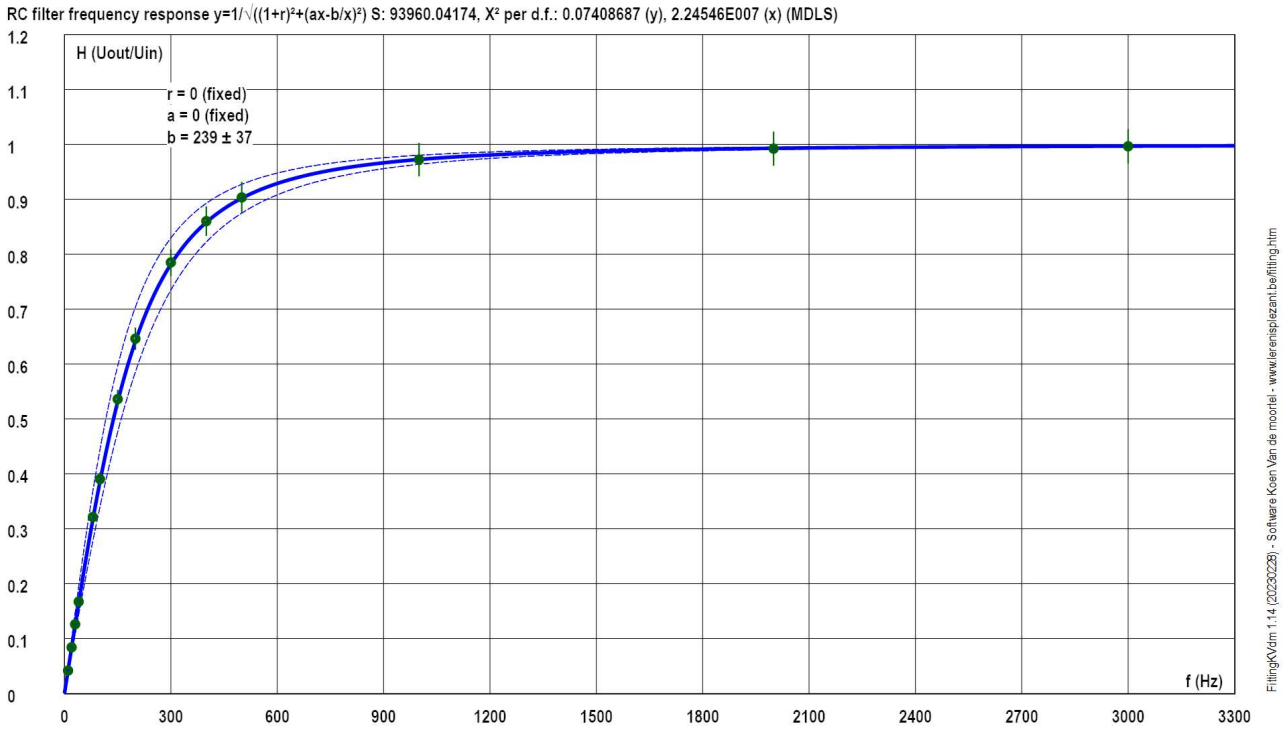


This fitting allows you to determine the value of L, without the need for an expensive multimeter. L = a·R/(2π) = 107.23mH (slightly different from the first measurement using the RLC filter and OLS).

* Filter 3 RC.dta1

Same setup as “Filter 1 RLC.dta1”, but without coil.

Using the model “RLC serial filter - preset RC”, you can determine the value of C if you don’t have a multimeter with this feature.



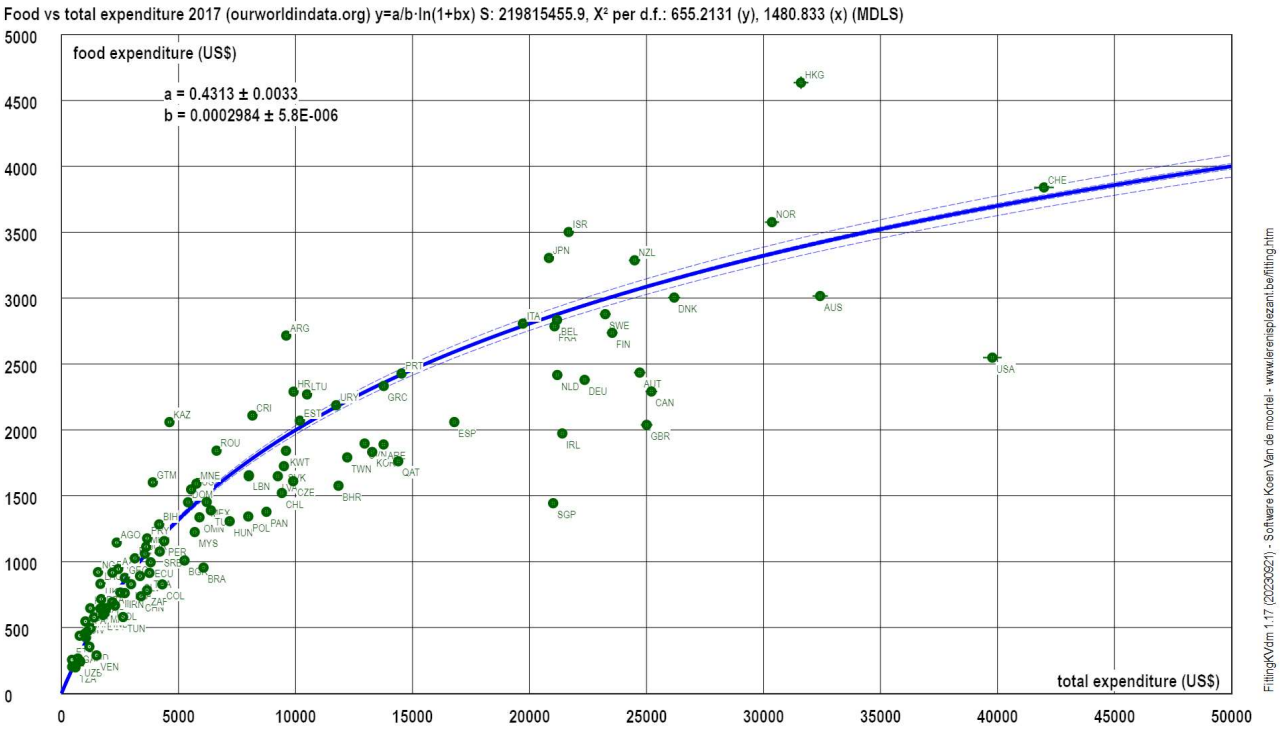
C = 1/(2πR·b) = 1.072µF, almost the same as in the first measurement (RLC, OLS).

* Food vs total expenditure 2017.dta1

Variables: x = total customer expenditure in US$ per person, 2017; y = expenditure for food (bought for consumption at home). Nothing about the error margins was known, so the errors were set proportional with the values (1%), which makes more sense here than setting the absolute errors the same.

Source: https://ourworldindata.org/grapher/food-expenditure-share-gdp

Since y can never exceed x, and rich people spend a smaller fraction of their income to food, a logarithmic function (shifted through the origin) seems a good model candidate.

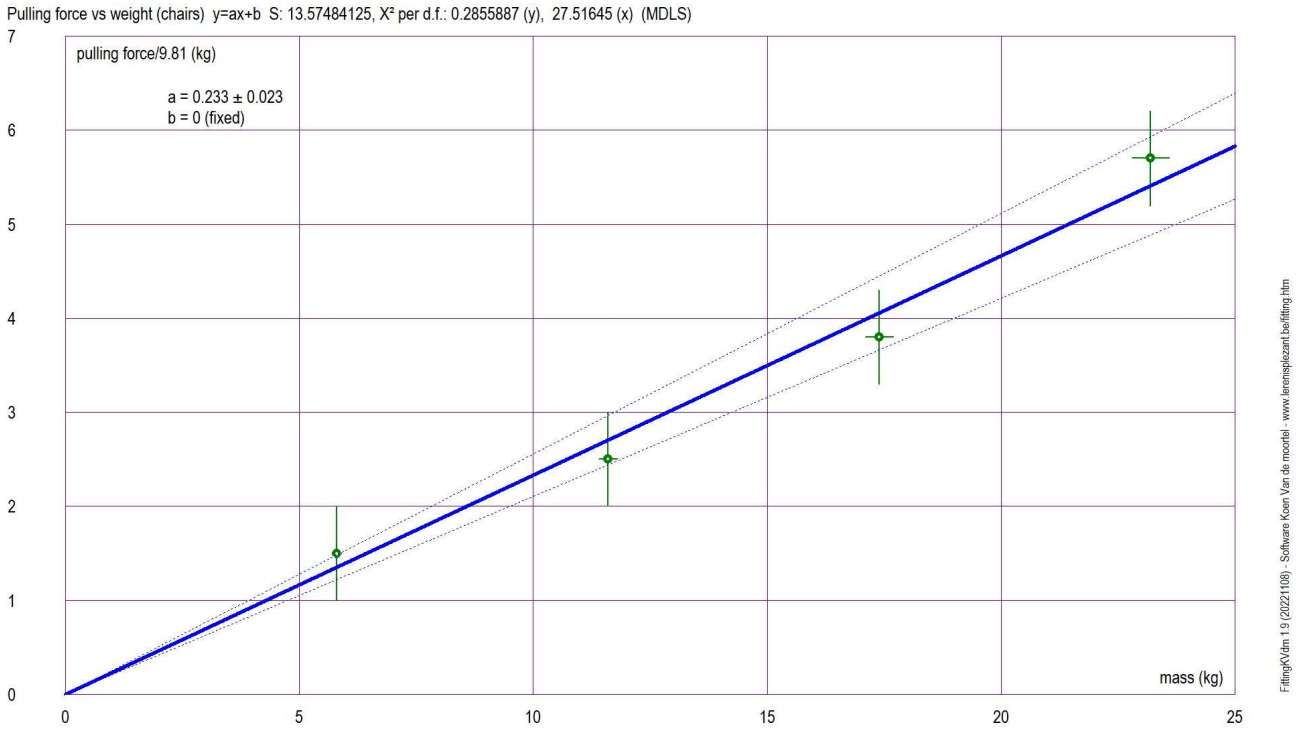


A power function seems to fit nice too, but that has the theoretical problem that it is always infinitely steep in the beginning, which means that y > x for a while.

* Friction of chairs on floor.dta1

The purpose of these measurements was to determine the **dynamic friction coefficient μ_D** of rubber on ceramic tiles. This was done by dragging chairs (with some kind of plastic (?) caps on their legs) over a floor.
Variables: x = mass of the dragged object (1, 2, 3 or 4 chairs on top of eachother) (in kg); y = the pulling force. Actually both were measured with the same dynamometer, calibrated in kg (a simple luggage scale), so to get the forces (pulling force and weight) in Newtons, we should multiply by 9.81, but that isn't necessary since μ_D is the ratio of both.
Only 1 chair was weighed, with a precision of ±0.1kg, so the error on the mass of n chairs was ±n·0.1kg. When doing this measurement, make sure the speed is as constant as possible (in that case: friction force = pulling force), so the reading will be constant too.

As expected, a linear relationship with b=0 fitted through the data within the error margins. The value obtained: $\mu_D \approx 0.233 \pm 0.023$, seems realistic.



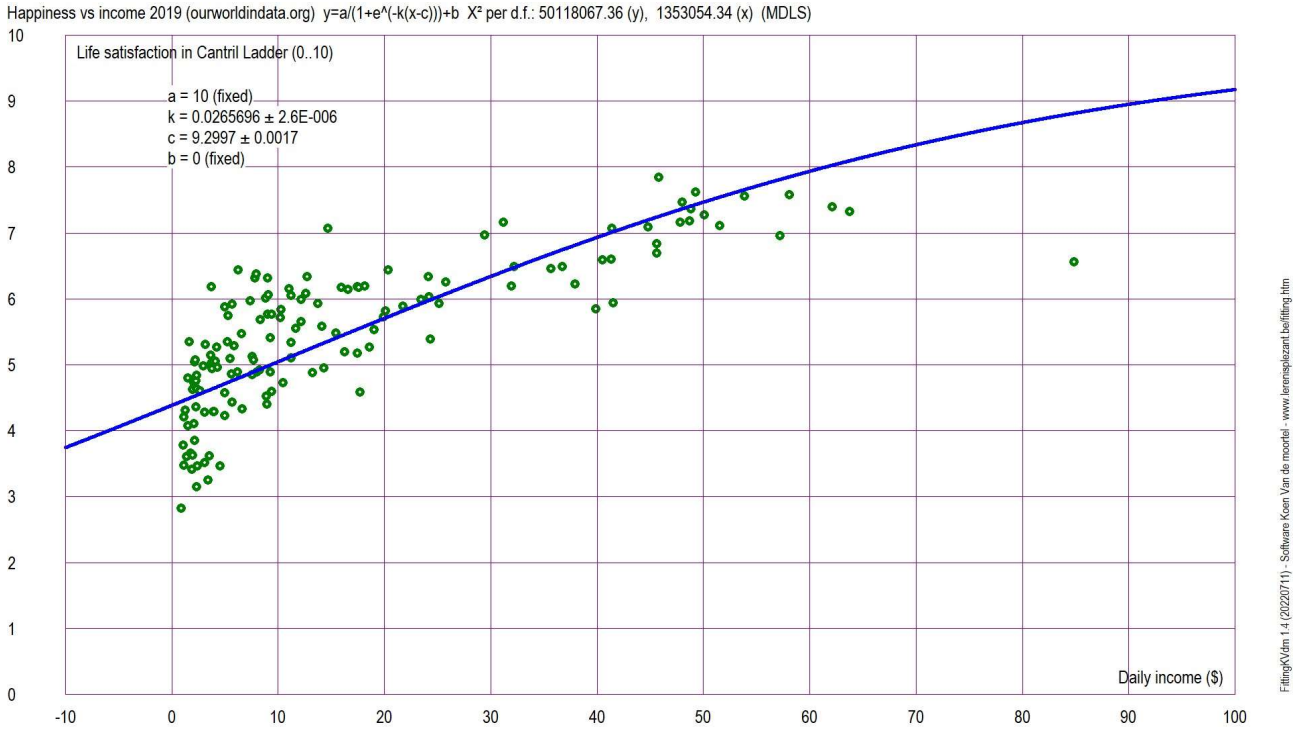
* **Happiness vs income.dta1**

Variables: x = median level of income or consumption per day ($), 142 countries, 2019, y = "Life satisfaction" in the Cantril Ladder scale (0..10). For some countries, one of the previous years was used.

Source:
World Happiness Report 2022
ourworldindata.org/happiness-and-life-satisfaction
ourworldindata.org/grapher/daily-median-income

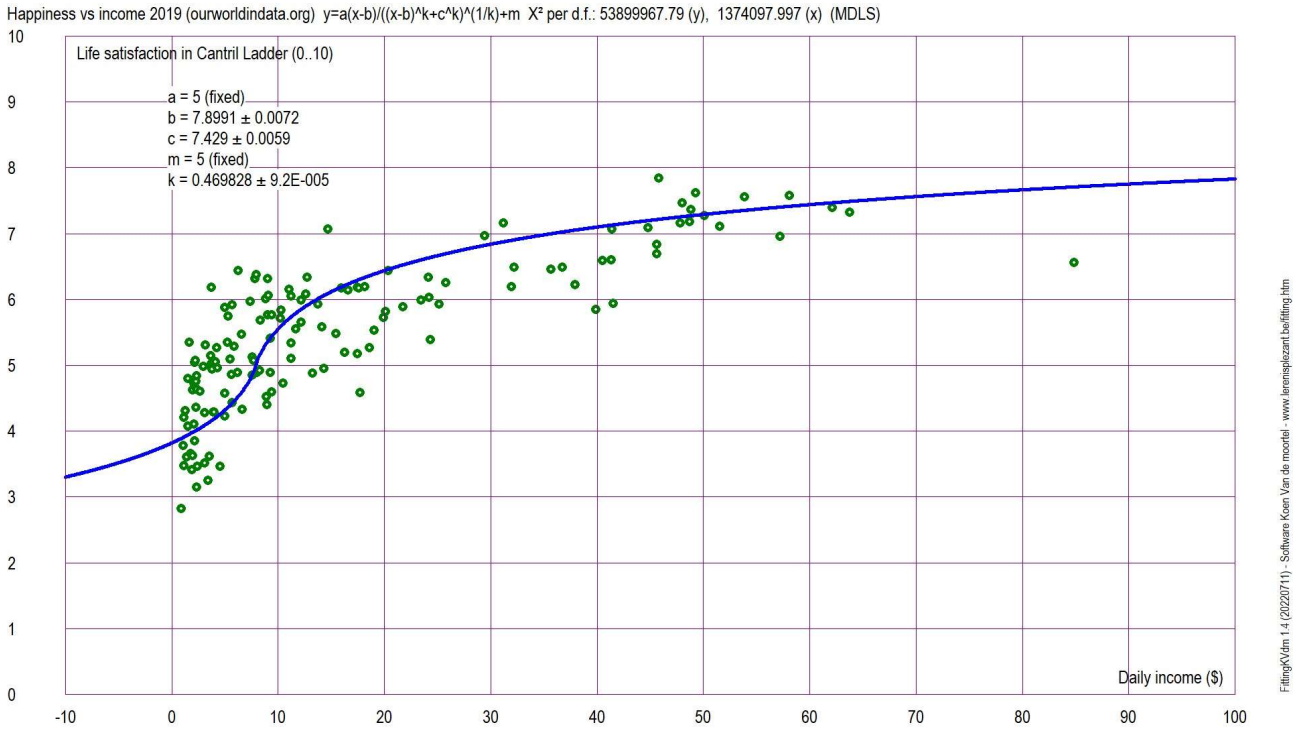
There is no physical law telling us what to expect of the relationship between money and happiness, so we have to experiment with models that make sense.

E.g. the logistic model with fixed parameters a = 10 and b = 0:

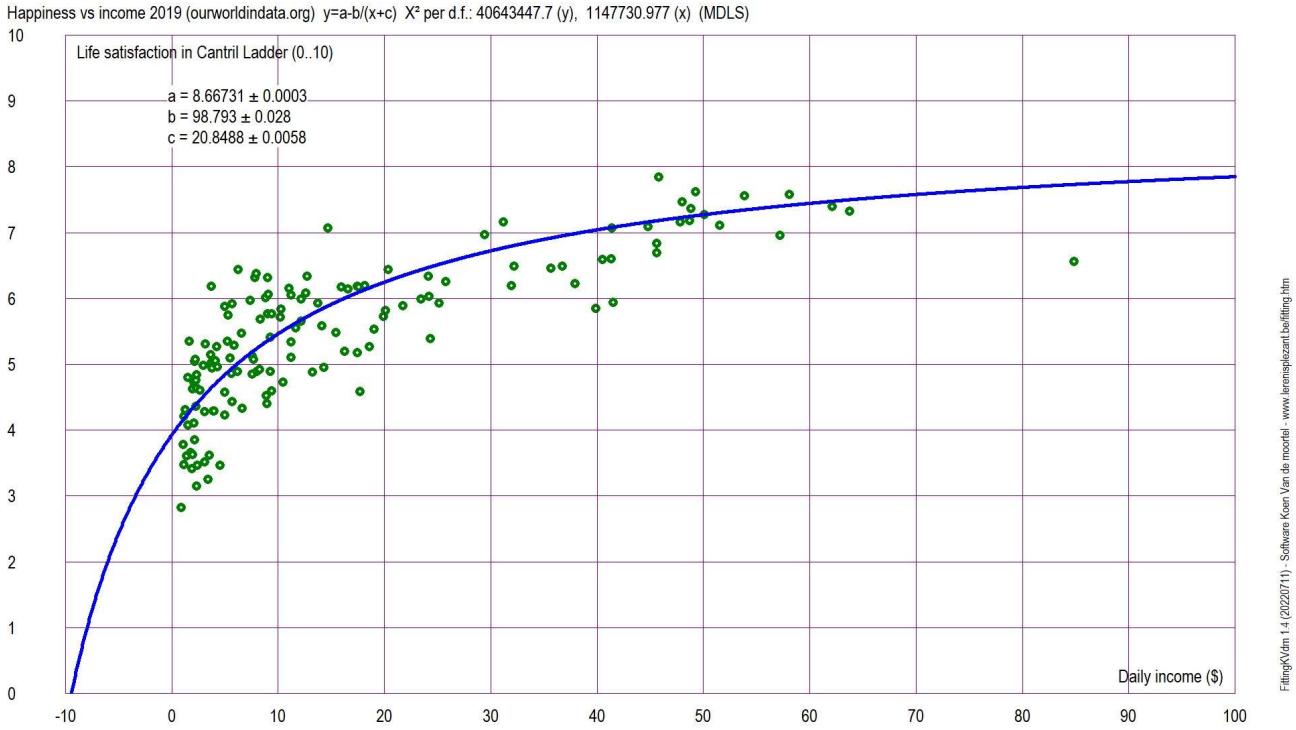


The happiness of the poorest countries seems to be overestimated here.

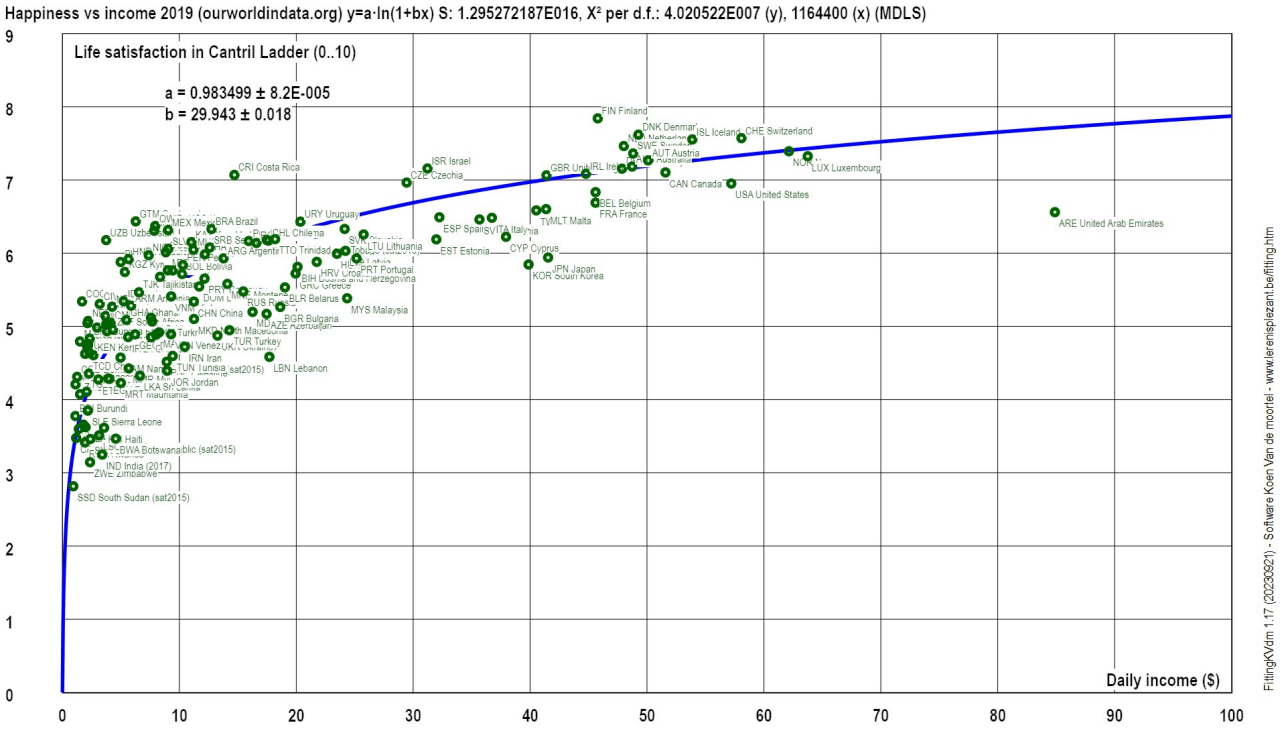
The "transition" model can be tried too. To obtain the same limits, we have to fix m = 5 and a = 5. This fits better with the poorer countries.



But, who says that everybody will be super happy if they only make enough money?
The "Rational 2" model with all parameters free, seems to fit well too. And it predicts an ultimate happiness average of 8.7 if everybody is very rich. Of course, the left part of the curve (negative income, to be interpreted as debts?) makes less sense.

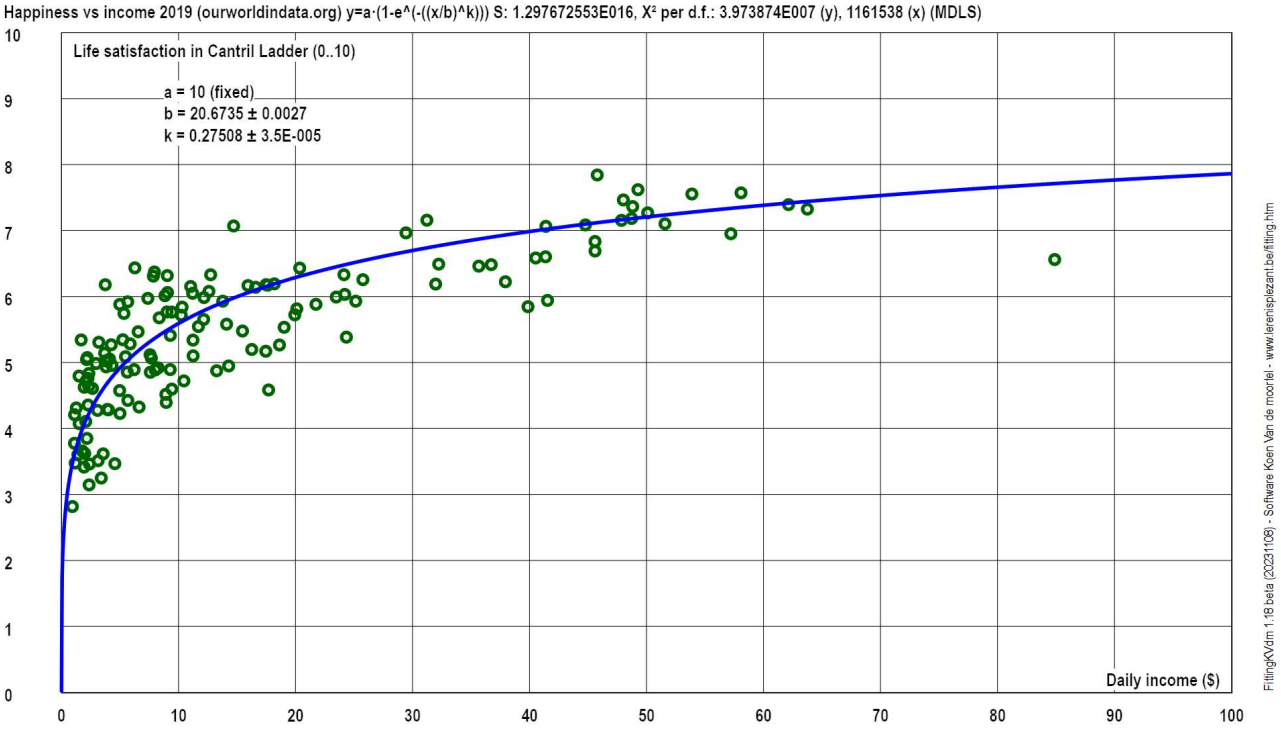


There is another model that seems to fit well: the "Logarithm shifted through 0", but then the assumption is: zero income = zero happiness…



The problem with this model is that it has no horizontal asymptote, so f(x) will eventually become more than 10 , which is impossible.

But… try the Weibull growth model! That is probably the best!



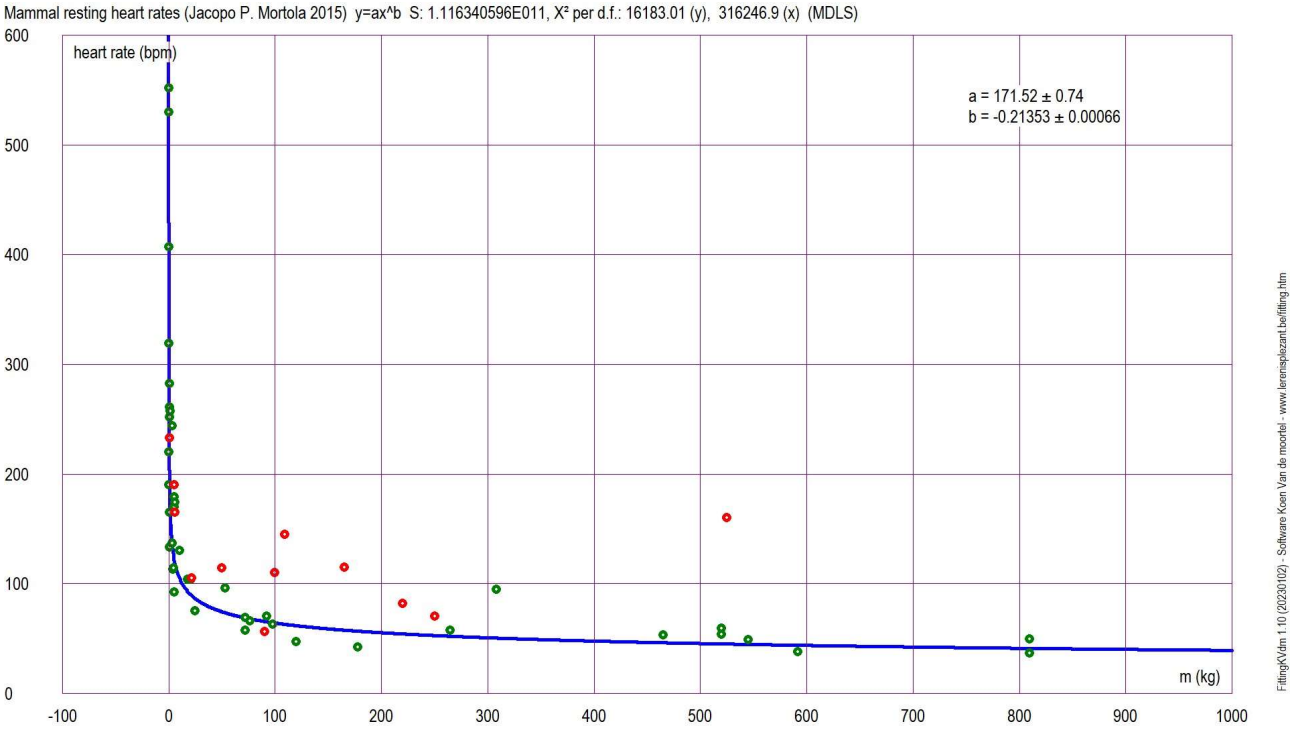
* Heart beat land mammals - Mortola 2015.dta1

Variables: x = mass of a (terrestrial) mammal (m, in kg); y = its average heart beat (in pulses per minute). The uncertainties on the values were not known, so we assumed them to be half of the last significant digit.

Source: Jacopo P. Mortola: "The heart rate - breathing rate relationship in aquatic mammals: A comparative analysis with terrestrial species", in: Current Zoology 61 (4): 569-577, 2015. https://academic.oup.com/cz/article/61/4/569/1803113

Since this is mainly a scaling relationship (like: big clocks have a lower resonance frequency), the "Power" model is probably the most appropriate.
Using MDLS we get an exponent of -0.21. OLS differs dramatically, unless the use of the x-uncertainties is switched off, but then we cheat, since the measurement errors are a reality. The culprit: the first to data point have a big weight since their x and y values are very precise, and by chance they happen to be "abnormal" (the second point has a bigger x and yet a bigger y), so the curve is pulled upwards enormously. MDLS however, knows how to deal with this!

Some of the dots are in red, why? Those are the measurements that were taken from *sedated* animals. It's quite obvious that that has an influence, so this seems a good reason to me to consider them as "outliers", so I set them "inactive".

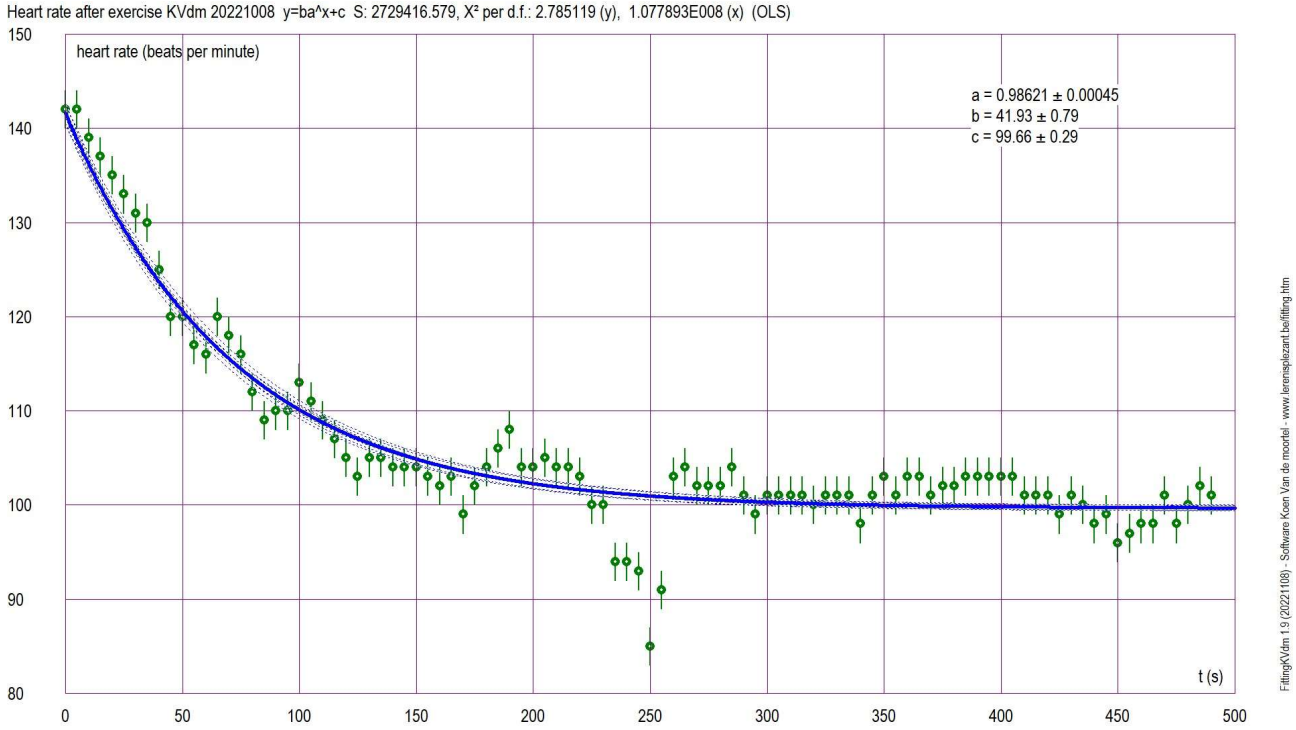


* Heart rate after exercise.dta1

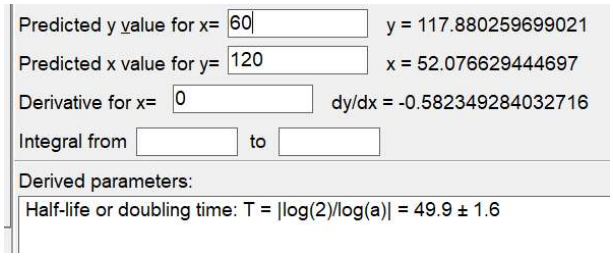
Variables: x = time (in s) starting from the end of a physical exercise; y = heart rate (beats per minute) measured with a simple sports watch with chest band (Kalenji Onrythm 110).

After an exercise, the heart rate will go back to normal after some time. This time is used as an indication for the condition of the person; the shorter the time, the better. See for example: https://www.ahajournals.org/doi/full/10.1161/JAHA.117.008341

This process of "going back to normal" can well be modeled with an exponential function:



This should give better results than individual measurements e.g. after 10 or 60s, because, as you see, there can easily be some "ups and downs" in the measurements (due to bad contacts of the sensor? or unknown physiological or psychological events, etc.). Fitting reduces this kind of errors maximally. Because of this "noise", OLS is recommended here above MDLS.



As an indicator for the **heart rate recovery** you could take several parameters:

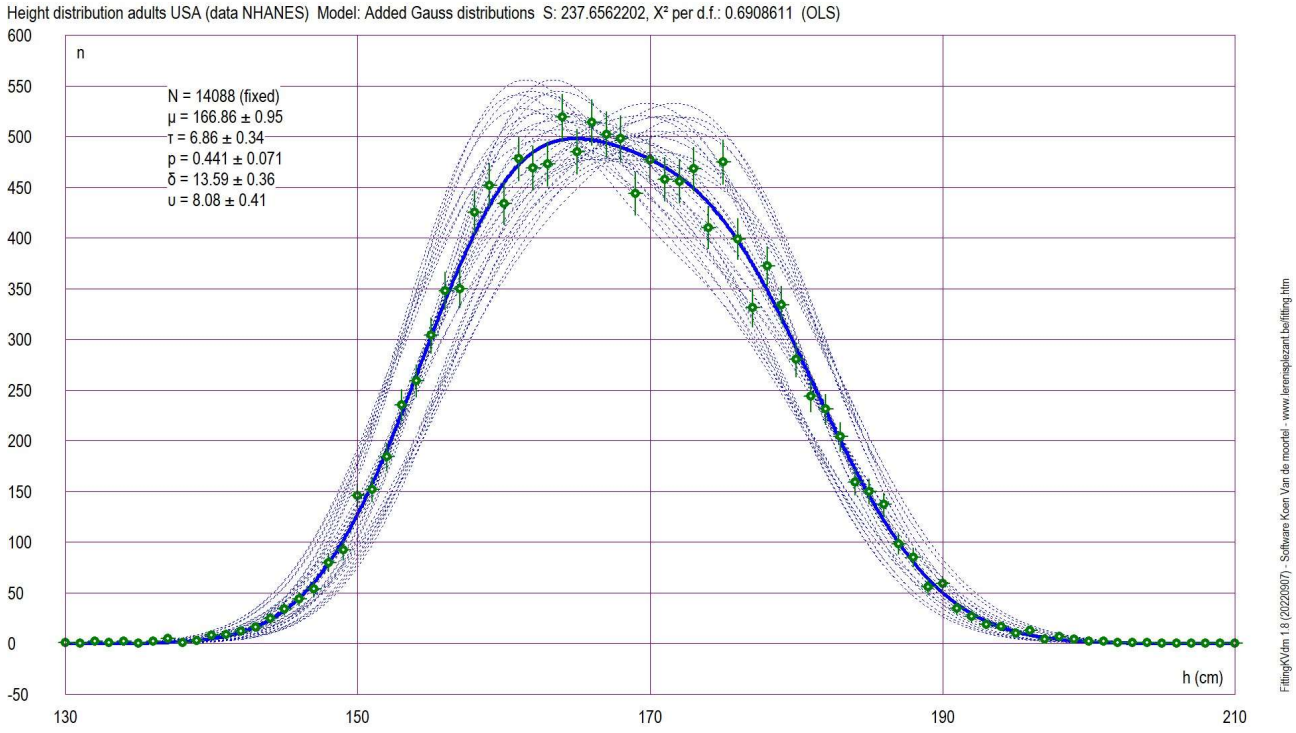
- the starting value minus the predicted value for x=60 (or whatever time span you want);
- the slope at t=0 (here -0.582);
- the time span needed to go back half the way to normal (here 49.9s);
- the parameter a (here 0.98621), which says how much the rate drops every second (here 1-0.98621 = 0.01379 = 1.379%).

* Height_distribution_adults_USA.dta1

Variables: x = height (h, in classes per cm); y = frequency of each class. The sample contains data from 14088 adults from the USA.

Source: NHANES

This looks like one peak, but the "Added Gauss" model will be able to distinguish two mixed groups (approx. the same size: 44.1±7.1% and 55.9±7.1% of the total) with averages that differ by 13.59cm, i.e. most probably the men and the women.

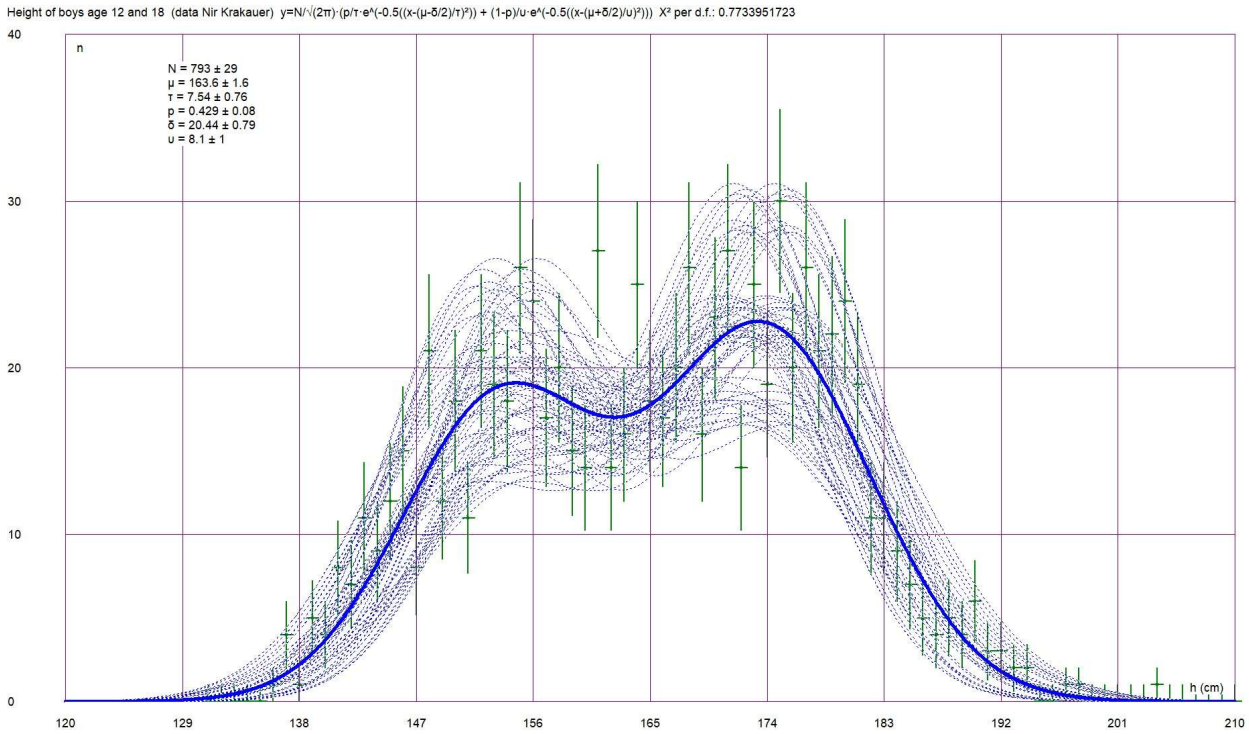


* Height_boys_12_18.dta1

Variables: x = height (h, in classes per cm); y = frequency of each class, in a mixed group of boys ages 12 and 18.

Source: NHANES

Try the “Added Gauss” model to find out how much boys grow on average between 12 and 18!

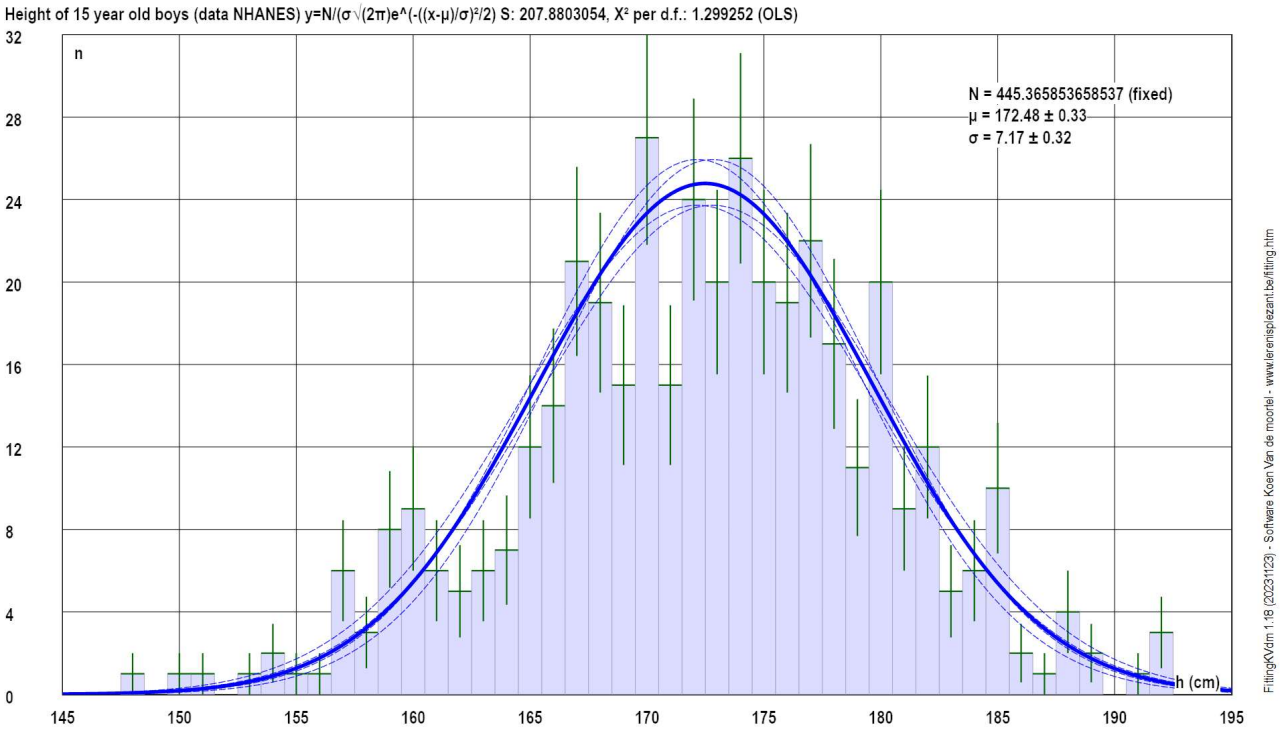


* Height_boys_15.dta1

Variables: x = height (h, in classes of 1cm); y = frequency of each class among 15 year old boys. For the uncertainties of the y values, the square root is used, because they are samples from a population. For the few y=0 values, an uncertainty of 1 was used; there seems to be no consensus about this practice, but it seems not implausible. See, e.g.: www.science20.com/quantum_diaries_survivor/those_deceiving_error_bars-85735

Source: NHANES

Try the "Gauss" model to find the average and standard deviation!



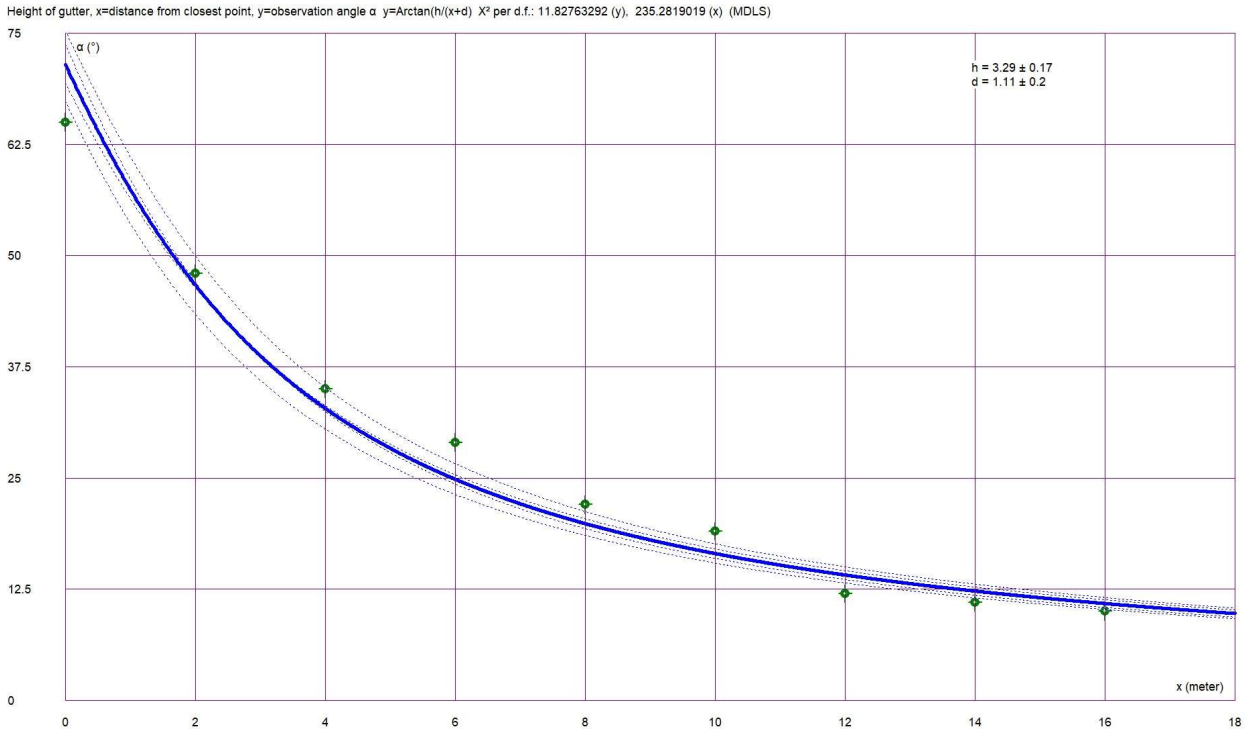
(Graph options: check "data bars" and set "size of data points" to 0.)

* Height_of_gutter.dta1

Variables: x = distance (x, in meters, measured from a closest point) to a tall object; y = angle of observation (α, in °) of a high mark on the object (in this case: the gutter of a house).
The purpose of the measurements is to determine the height of the mark (above the eyes of the observer).

The “Parallax” model was designed for this!

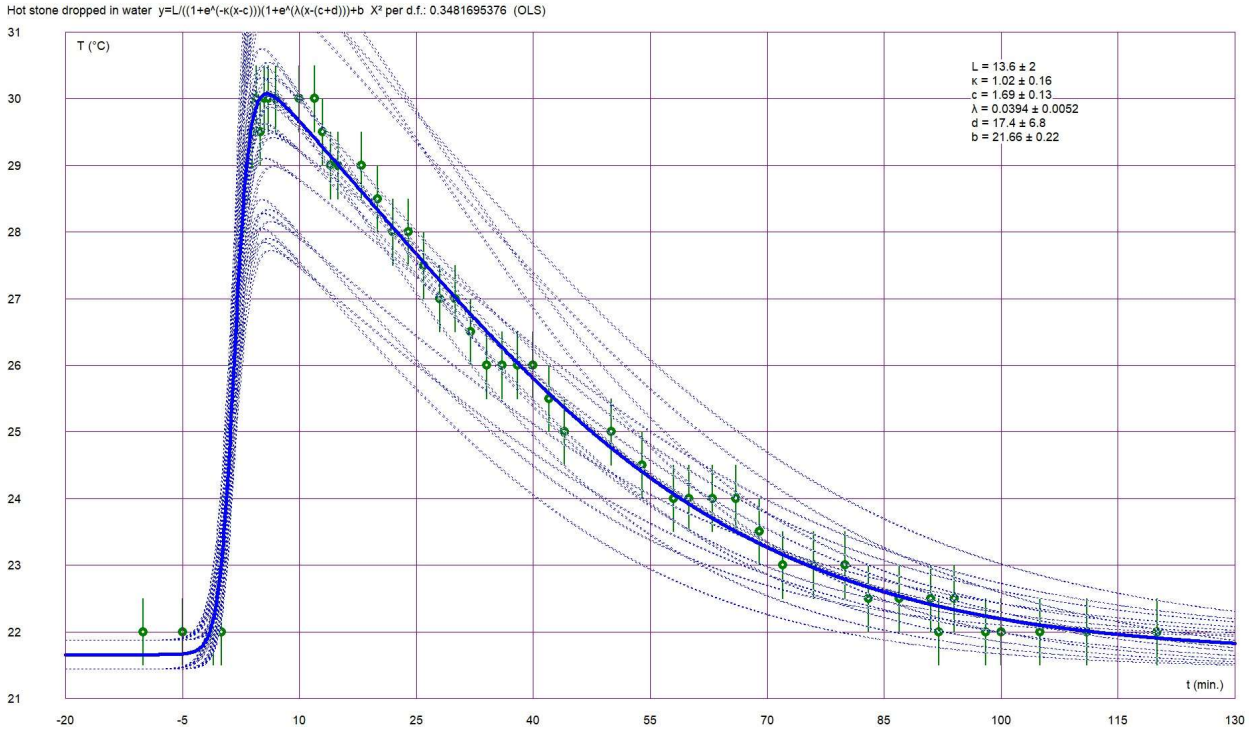
Remark: you might notice that the “exponential” model fits nicely too, but… you will see that it makes no sense if you extrapolate to the left: the curve keeps going up nicely even if you go to negative x values, while in reality the angle can never exceed 90°!



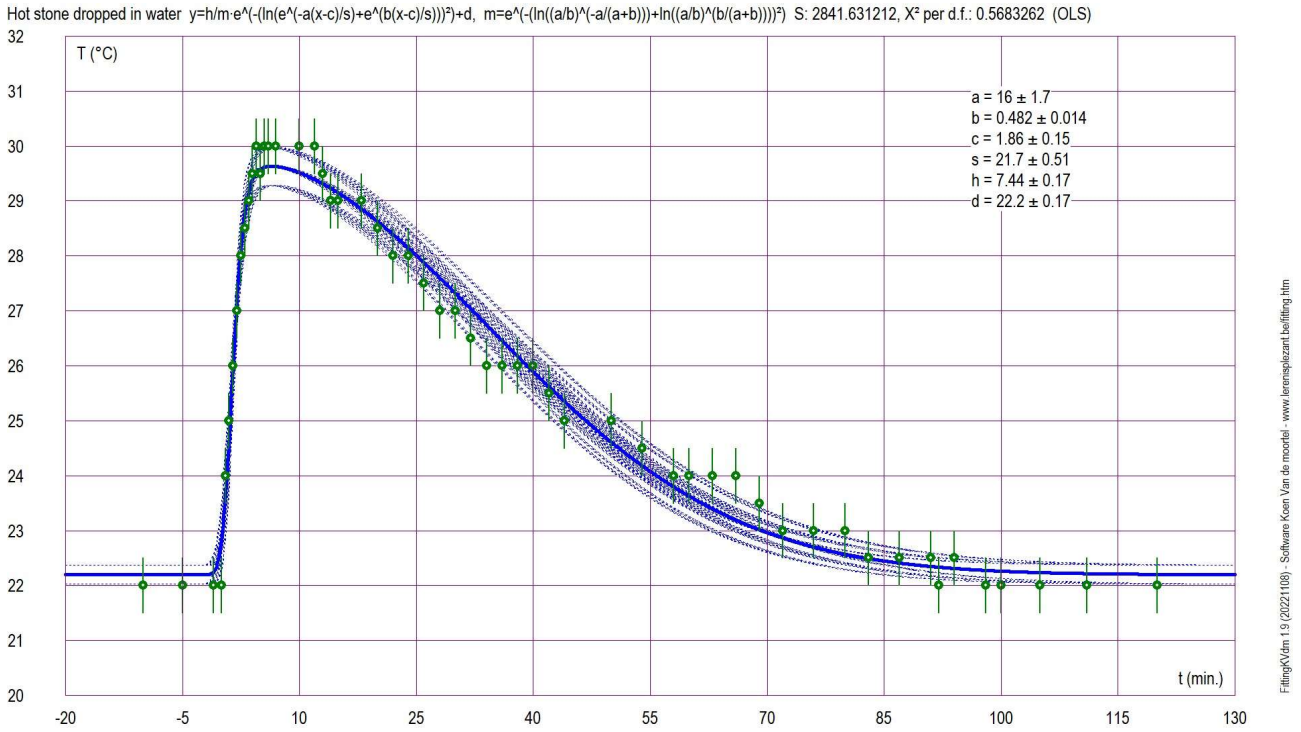
* Hot_stone_water.dta1

Variables: x = time (t, in minutes, starting when a hot stone is dropped in a bowl of water at room temperature); y = Temperature of the water (T, in °C).

Try the “Double Logistic + baseline” model to see the pattern.



The “skewed peak 2” model even seems to fit better:



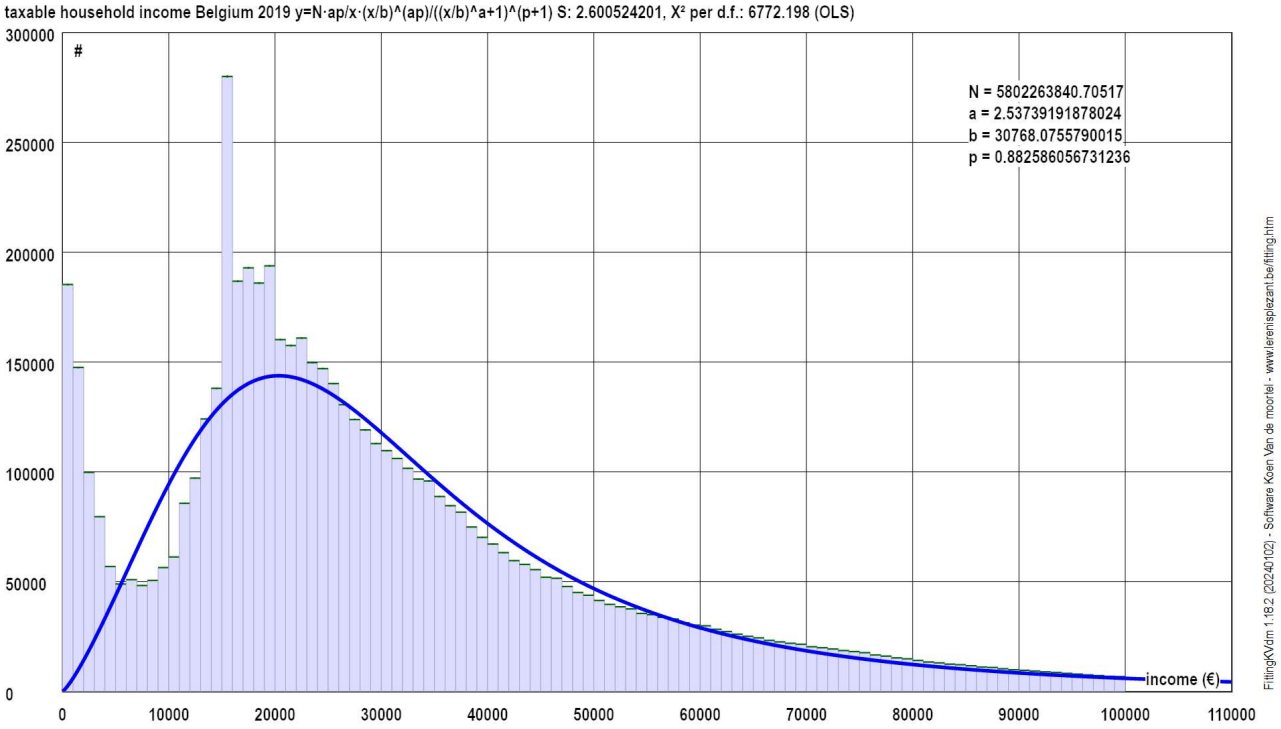
* **Income distribution Belgium 2019.dta1**
* **Income distribution of households USA 2020.dta1**

Variables:
x = income class, y = number of people in that class.

Sources:
statbel.fgov.be/nl/themas/huishoudens/fiscale-inkomens/plus
www.census.gov/data/tables/time-series/demo/income-poverty/cps-hinc/hinc-06.html

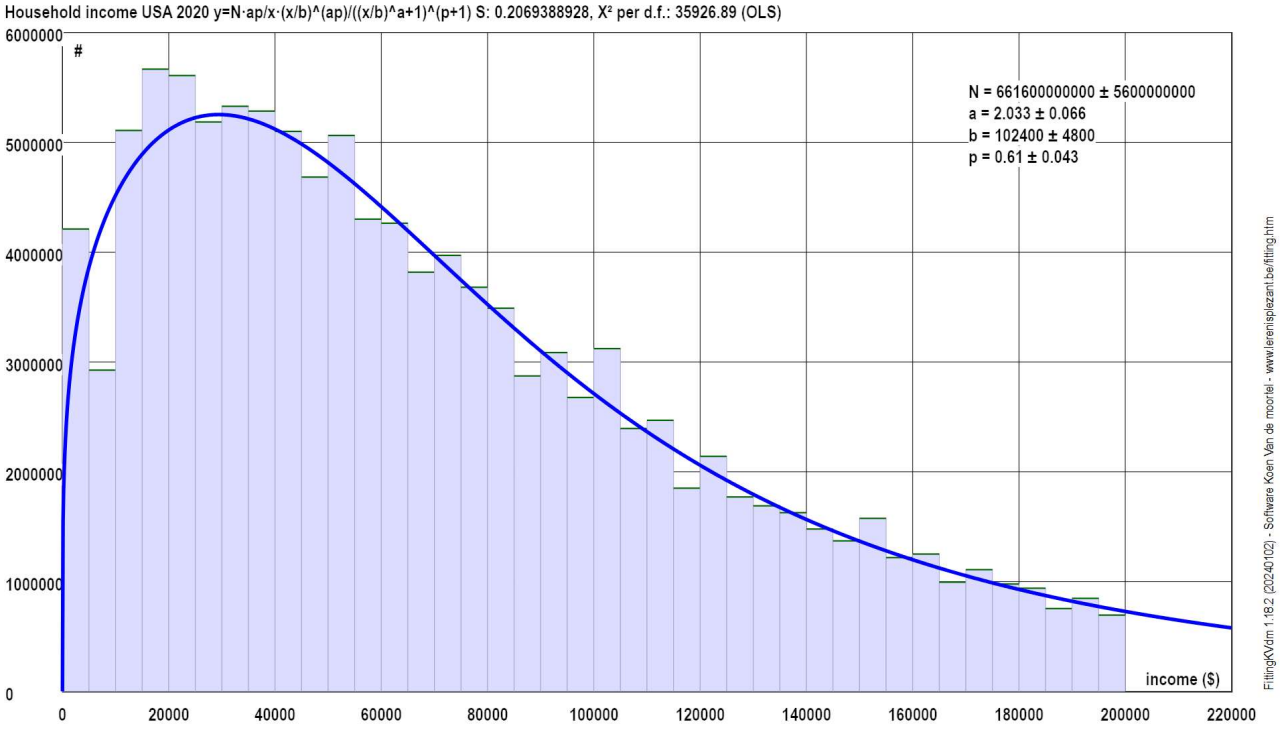
As expected, these data don't look normally distributed. They look more like a mixture of different skewed peaks, especially the belgian data, since there are employed people (with minimum wage), unemployed people (usually receiving about 800...1300€ monthly), self-employed people (who often struggle to survive and might even have negative incomes, but those are hidden from the statistics), retired people,...

A Dagum distribution with p = 2 fits reasonably with the right tail of the belgian data, but of course not with the left side:



(Choose the graph option "data bars" to get the graph in this form.)

The American data fit better, but on the left side is also something wrong:

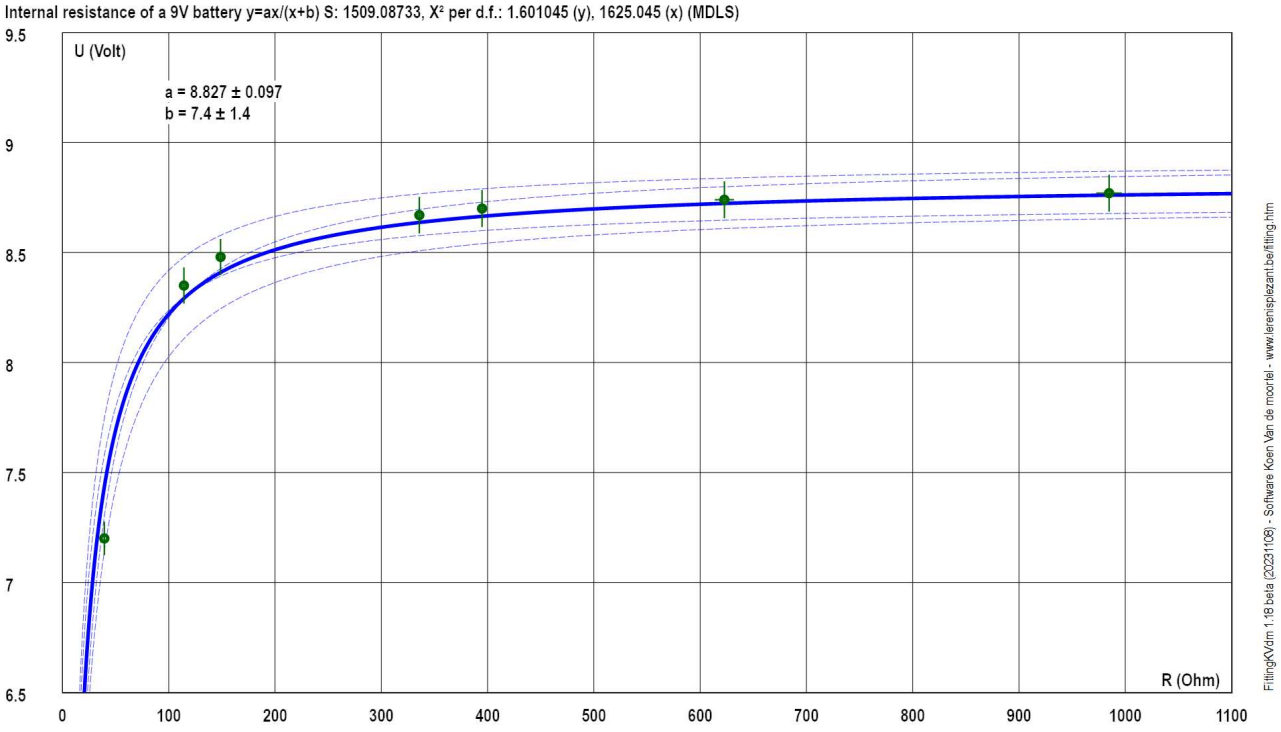


* Internal_resistance_9V_battery.dta1

Variables: x = load resistance (R) parallel with a 9V battery, y = Voltage (U) over the resistor.

The smaller R, the more U drops due to the internal resistance of the battery R_b. Theoretically $U = U_0 \cdot R/(R+R_b)$ with U_0 the voltage if no resistor is attached ($R = \infty$), so the "Rational 1" model is appropriate to determine R_b (= parameter b).

Warning: the last two voltage measurements were the same (8.81) due to the lack of precision of the voltmeter, which makes it a bit harder for the MDLS algorithm to converge, but it will work if you keep the lower limit for parameter a (= U_0) at 8.81. Notice that the OLS algorithm will predict 9.029 for the final voltage, which is just impossible! MDLS predicts a very plausible 8.827!
The predicted values for the internal resistance differ significantly too:
MDLS: $R_b = 7.4 \pm 1.4\Omega$, OLS: $R_b = 9.97 \pm 0.78\Omega$.

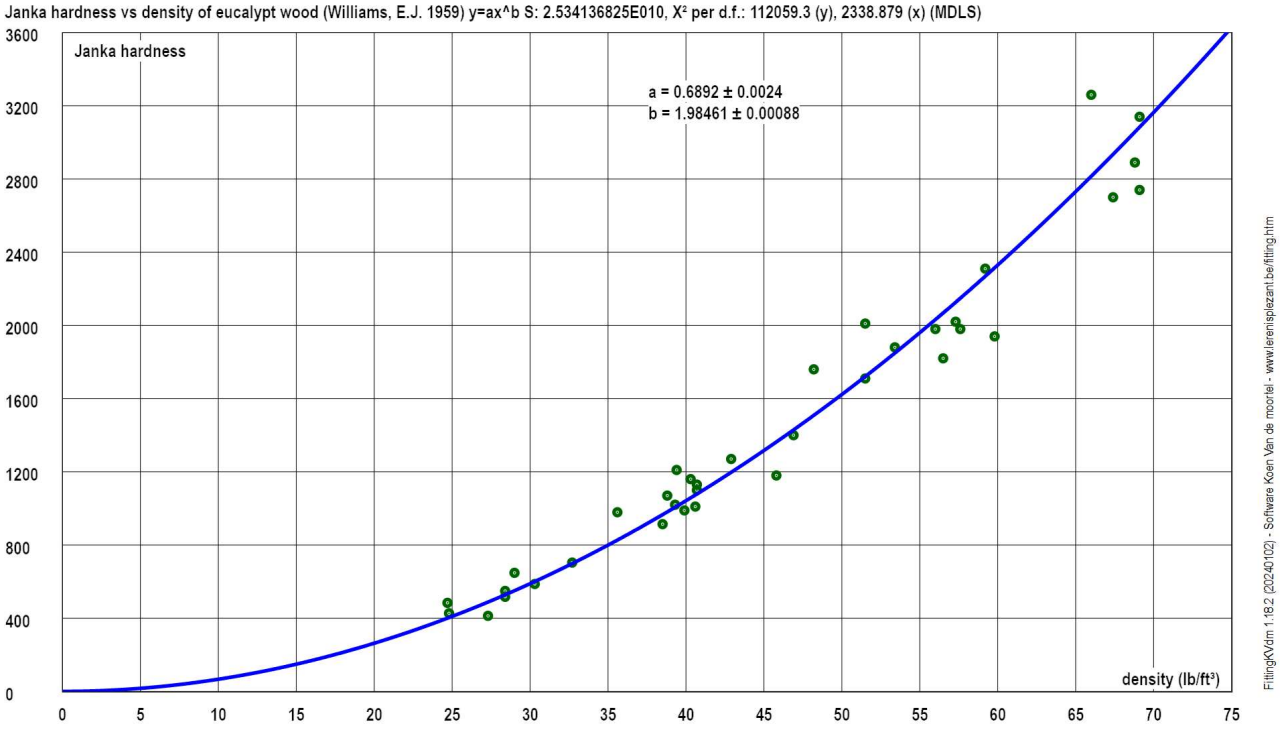


* Janka hardness vs density of eucalypt wood.dta1

Variables: x = density (in the old Anglo-saxon unit system: lb/ft³) of Australian Eucalyptus wood types; y = hardness on the scale of Janka, see:
https://en.wikipedia.org/wiki/Janka_hardness_test

Source: E.J. Williams (1959), quoted as example 334 in "A Handbook of Small Data Sets", by D.J. Hand, F. Daly, A.D. Lunn, K.J. McConway and E. Ostrowski, Springer 1994.

A power function fits the relationship reasonably well:

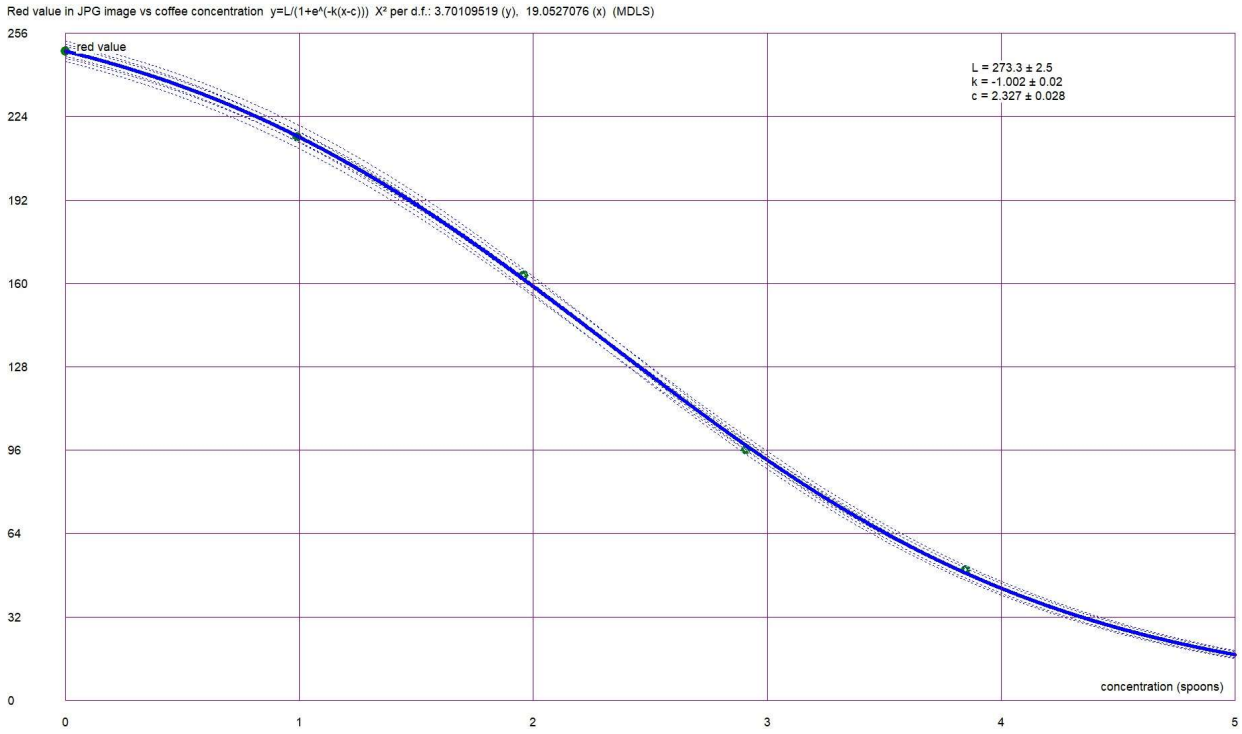


* **JPG-Red_vs_coffee_concentration.dta1**
* **JPG-Green_vs_coffee_concentration.dta1**
* **JPG-Blue_vs_coffee_concentration.dta1**

Variables: x = coffee concentration (c), in spoons from a strong reference mix, in a full glass of water; y = red/green/blue value (0..256) in a JPG image taken with a mobile phone camera from the solution with indirect sunlight going through the glass.

The more spoons of coffee powder in a bottle, the darker it becomes. But the tone mapping algorithm in the camera influences the curve.

Try the "Logistic" model (with fixed b = 0) to calibrate the camera as a concentration measuring device!



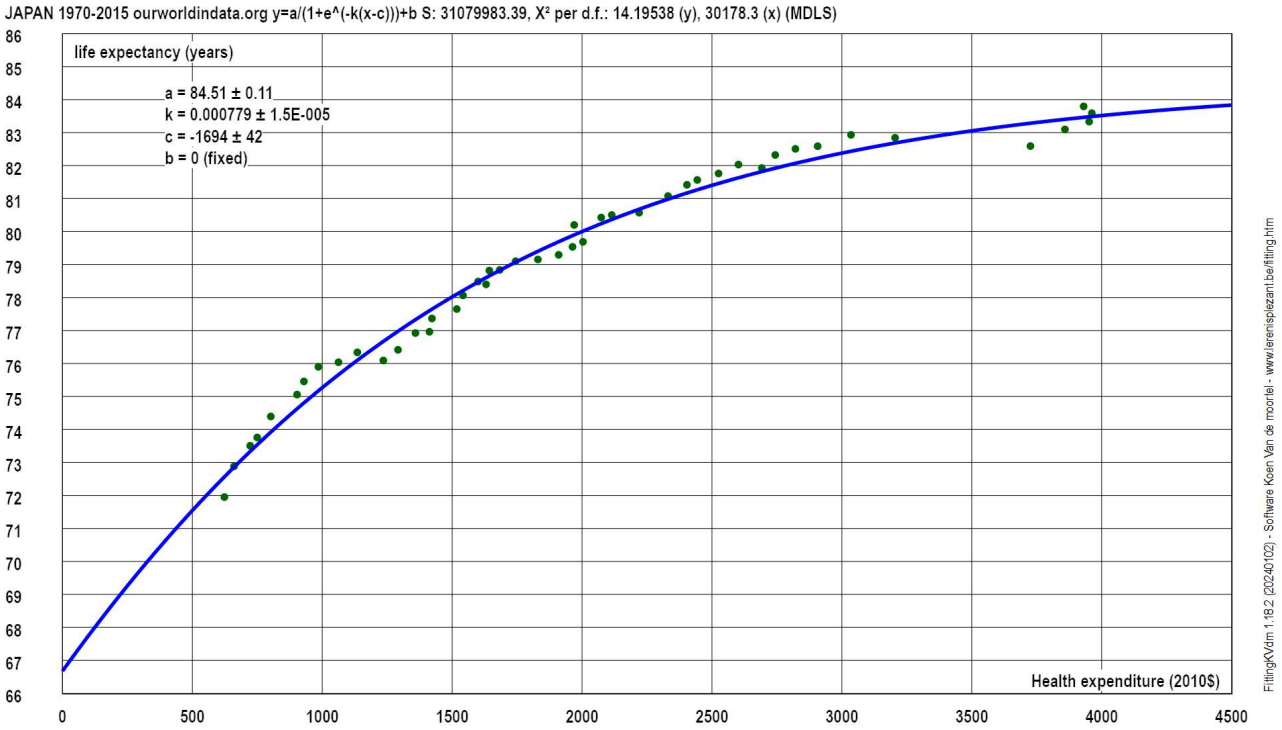
* **Life exp vs health spending CH.dta1**
* **Life exp vs health spending JAPAN.dta1**
* **Life exp vs health spending USA.dta1**

Variables: x = health expenditure per capita (2010 int.-$), y = life expectancy (years), for Switzerland, Japan and USA 1970-2015.

Source of the data:
ourworldindata.org/the-link-between-life-expectancy-and-health-spending-us-focus

The fit with the "Rational 2" model is quite good, but so are the logistic (baseline 0) & transition models, and the "plateau-exponential" and the Gompertz growth (baseline 0). You could try other models like linear and quadratic, but you will see how stupid that is if you look at their predictions for x going to infinity!

The models that make sense predict an average life of about 58 to 67 years without medical costs (= f(0)), and about 84 to 89 with unlimited expenses, which makes more or less sense.

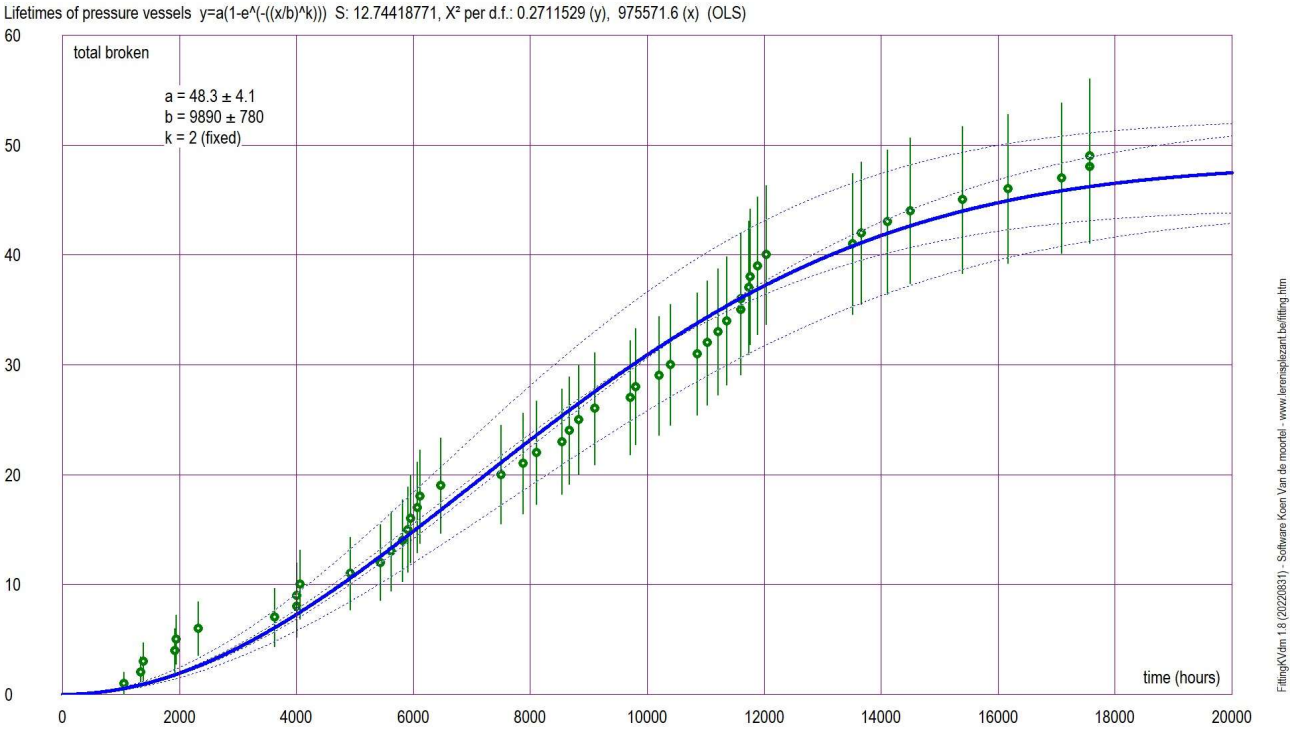


* Lifetimes_pressure_vessels.dta1

Variables:
The original data were 49 ages of pressure vessels when they broke down (in hours). In the source text, a distribution was calculated by grouping them in 7 classes. Then a Weibull distribution was fitted through these 7 numbers. However, doing this, a lot of information is lost. What I did, conserves all the information: x_1 = the time when the first vessel broke, y_1 = 1; at time x_2, the second vessel broke, y_2 = 2, and so on: y_i = the total number of broken vessels. I assumed the same error (1 hour) for each time value. For the y values I used the usual square root. Now we have to fit with the cumulative distribution of course (the "Weibull growth" model).

Source:
"An introduction to generalized linear models", 2nd ed., by Annette J. Dobson, Chapman & Hall/CRC, 2001, chapter 4.2: Example: Failure times for pressure vessels.
This can be found here:
www.academia.edu/18454337/AN_INTRODUCTION_TO_GENERALIZED_LINEAR_MODELS

The text above recommends to use the Weibull distribution with k = 2.
Since the last data points are probably in the horizontal asymptote area, it might be safer to use OLS here. MDLS produces a slightly higher value for parameter a.

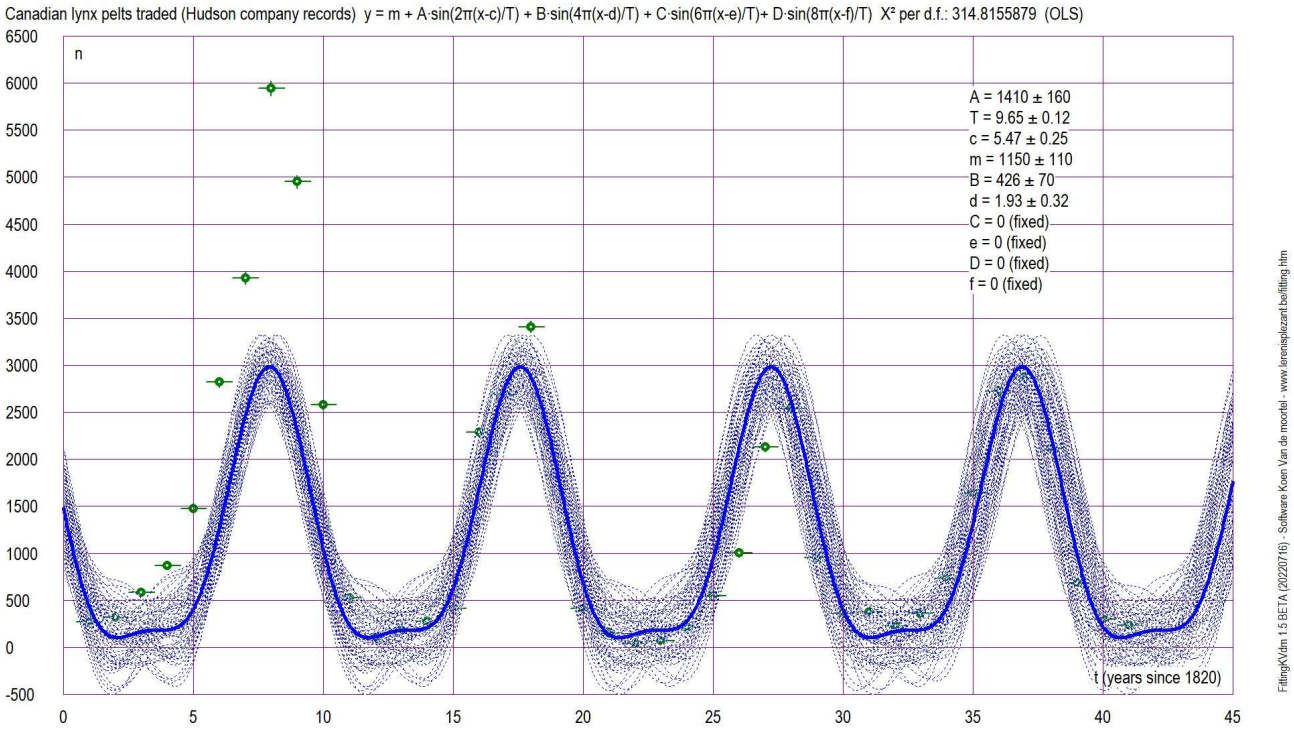


* Lynx.dta1

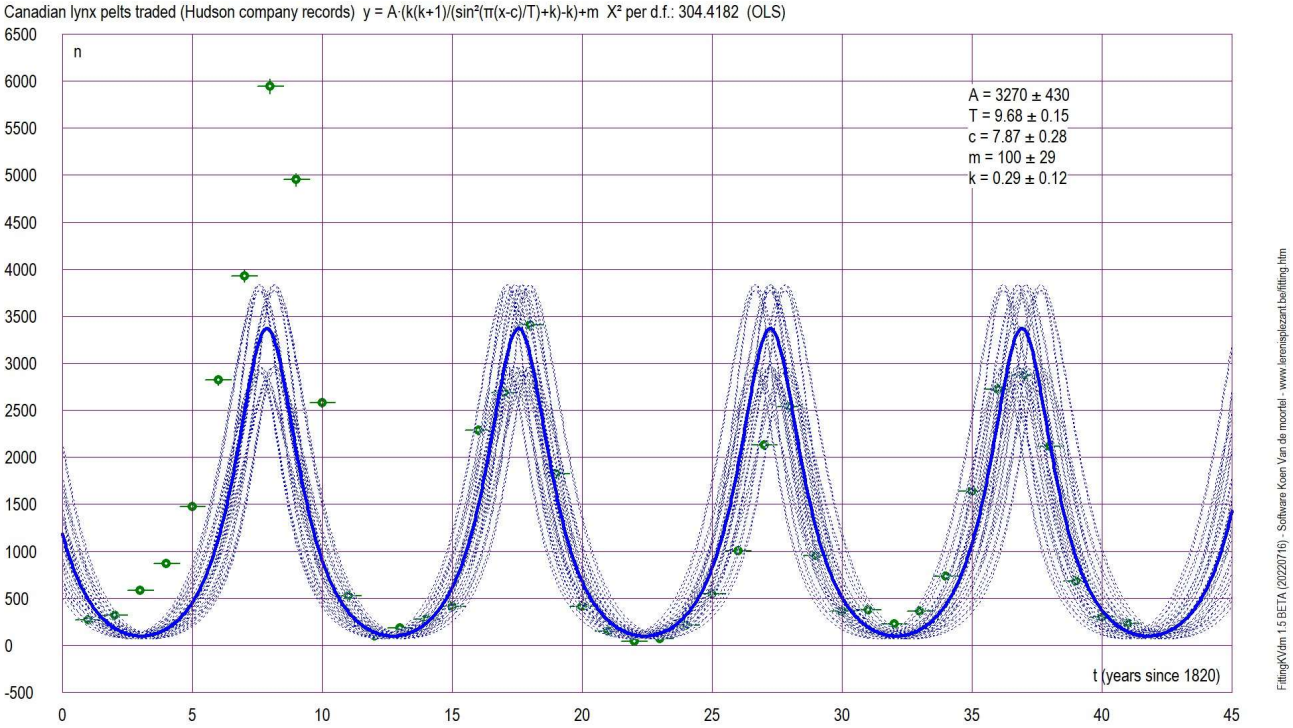
Variables: x = time (years since 1820), y = number of Canadian lynx pelts traded by the Hudson company. If we use this number as an indicator for the total lynx population, we can use the square root of each number as an estimation of the uncertainty.

Source:
Charles Elton and Mary Nicholson: "The Ten-Year Cycle in Numbers of the Lynx in Canada." (Journal of Animal Ecology, vol. 11, no. 2, 1942, pp. 215–44. JSTOR, https://doi.org/10.2307/1358). It can be downloaded here:
jxshix.people.wm.edu/2009-harbin-course/classic/Elton-1942-J-Anim-Ecol.pdf

The "Sine wave with harmonics" model with just the main wave and the first harmonic (half period) might serve reasonably to model these data:



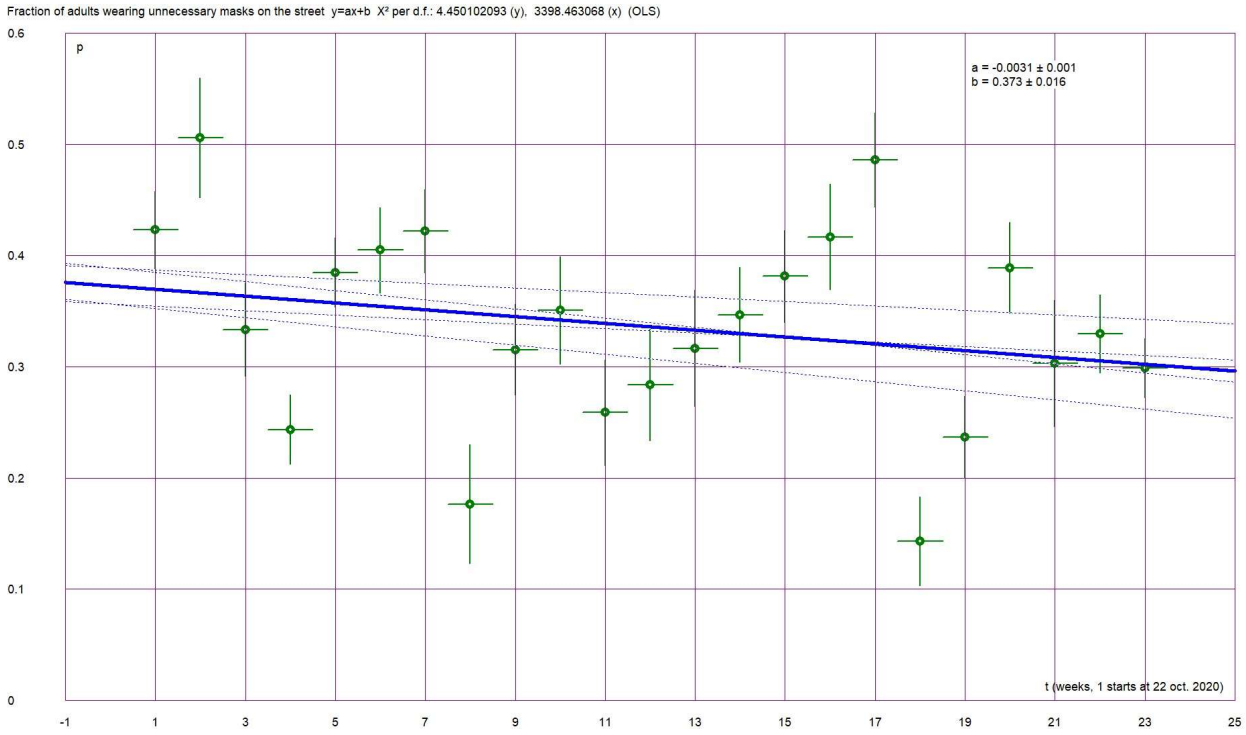
But also the "Periodic peaks" model fits reasonably. The difference is that the peaks are symmetrical and the curve is a bit flatter between the peaks.


Canadian lynx pelts traded (Hudson company records) y = A·(k(k+1)/(sin²(π(x-c)/T)+k)-k)+m X² per d.f.: 304.4182 (OLS)
n
A = 3270 ± 430
T = 9.68 ± 0.15
c = 7.87 ± 0.28
m = 100 ± 29
k = 0.29 ± 0.12
t (years since 1820)
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
-500
0
5
10
15
20
25
30
35
40
45
FittingKVdm 1.5 BETA (20220716) - Software Koen Van de moortel - www.lerenisplezant.be/fitting.htm

* Masks.dta1

Variables: x = time in weeks starting from 22 oct. 2020; y = the fraction (p) of adults wearing masks on the street (where it wasn't obligatory, in my neighborhood).
The data were grouped per week to have enough observations (n) to calculate the $\sigma = \sqrt{(p \cdot (1-p)/n)}$.

Try the "Constant" model to calculate the weighted average and its standard deviation. You might look for a trend by using the "Linear" model, but extrapolating doesn't make much sense, since there is no obvious causal relationship. That's also why MDLS doesn't make sense here.

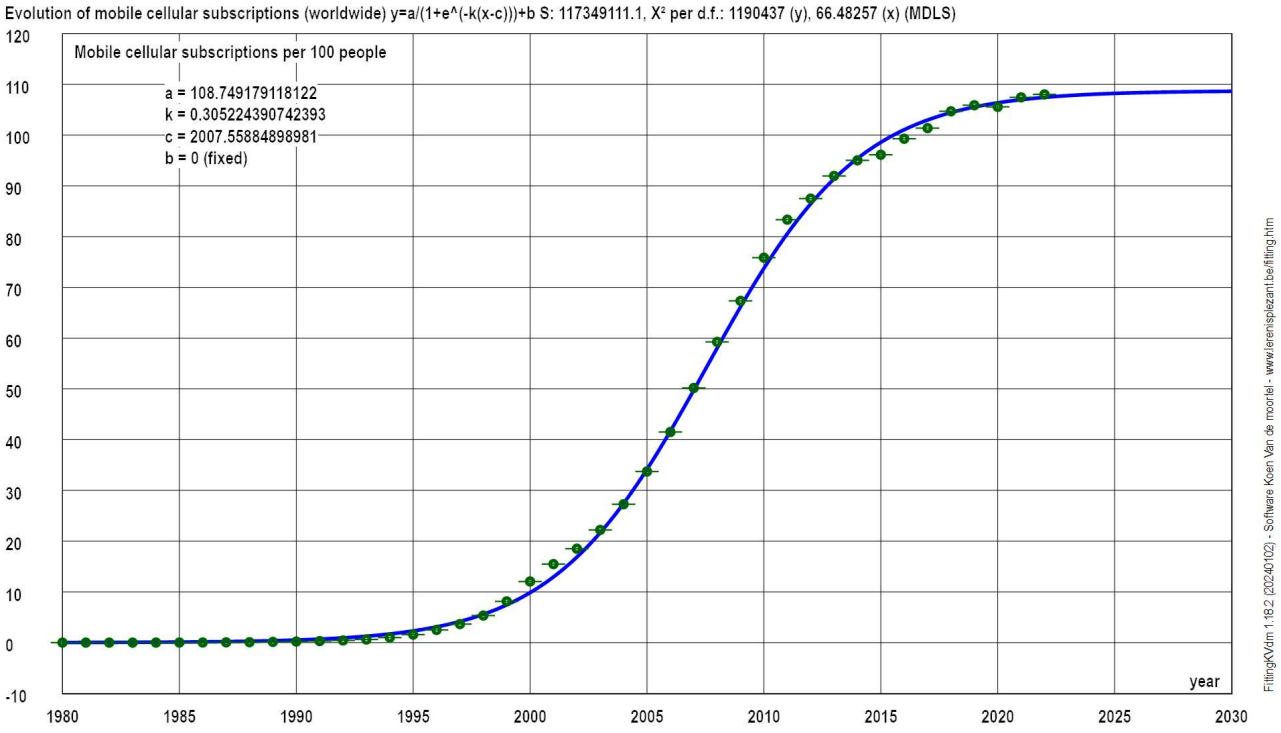


* Mobile phone usage.dta1

Variables: x = time (calendar year), y = mobile cellular subscriptions per 100 people; the uncertainty on the y data was unknown (set to the same value 0.001).

Source: International Telecommunication Union (ITU) World Telecommunication/ICT Indicators Database (https://data.worldbank.org/indicator/IT.CEL.SETS)

Since we might assume this is an example of a transition from zero to a certain "saturation" value, a logistic model (with b = 0) is plausible.



If this model reflects the reality, we are now (2024) near this saturation level.

The "Transition" model also fits reasonably but you can't force that to start from zero. "Gompertz growth" fits clearly worse.

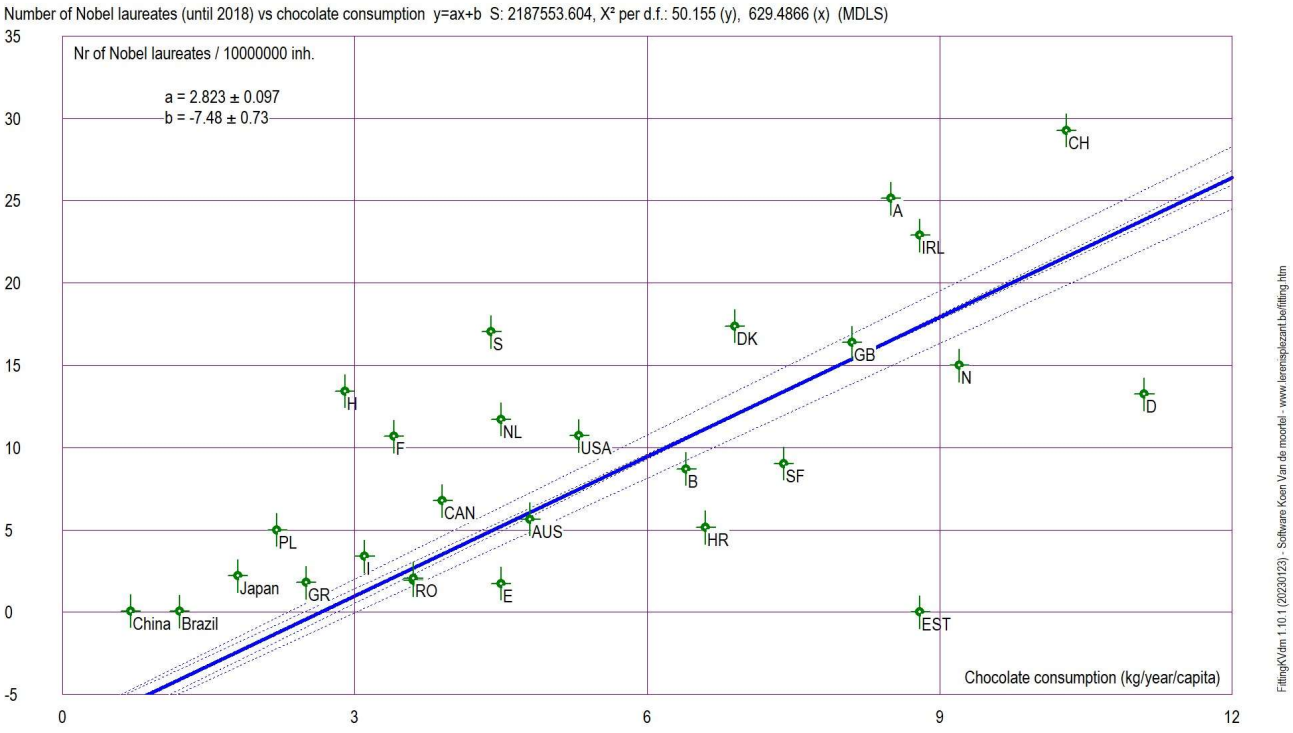
*** Noble prizes and chocolate.dta1**

Variables: x = chocolate consumption (kg/year/capita); y = number of Nobel laureates (up to the year 2018) per 10000000 inhabitants, for 27 countries.

Source: Aloys Leo Prinz: "Chocolate consumption and Noble laureates", Social Sciences & Humanities Open, Volume 2, Issue 1, 2020, 100082
www.sciencedirect.com/science/article/pii/S2590291120300711#bib22
en.wikipedia.org/wiki/List_of_countries_by_Nobel_laureates_per_capita

Eating chocolate seems to be good for your intelligence! Seriously? I quote: "It remains unclear whether the correlation is spurious or an indication for hidden variables."

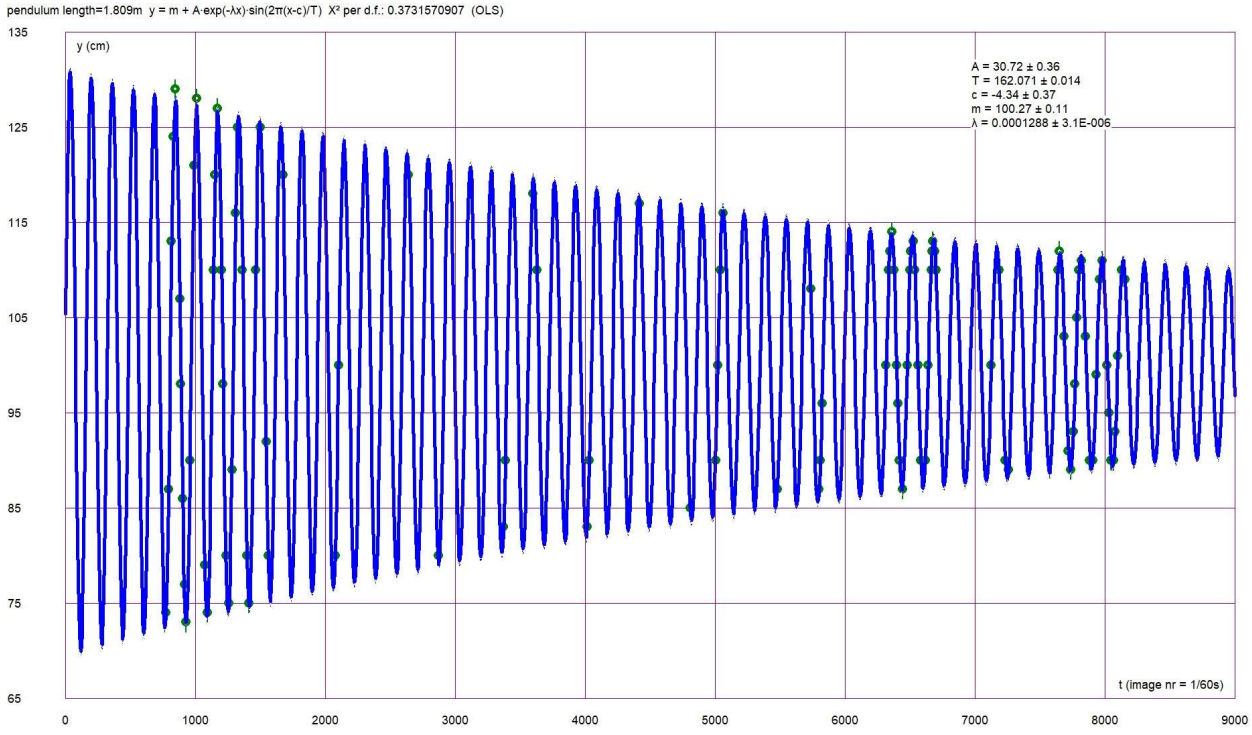
Anyway, you can try with a linear model (MDLS and OLS will differ a lot since the data points are more forming a cloud than a straight line).



* Pendulum.dta1

Variables: x = time in image nrs (1/60s); y = horizontal position (in cm) of a swinging object on a rope, the equilibrium position was 100cm on my ruler.

From the physical theory, the “Damped sine wave” model will give the best fit, of course. A good starting value for the period can be derived from the consecutive passages through the equilibrium position around t=6500, but the “Guess” button will do it easier!



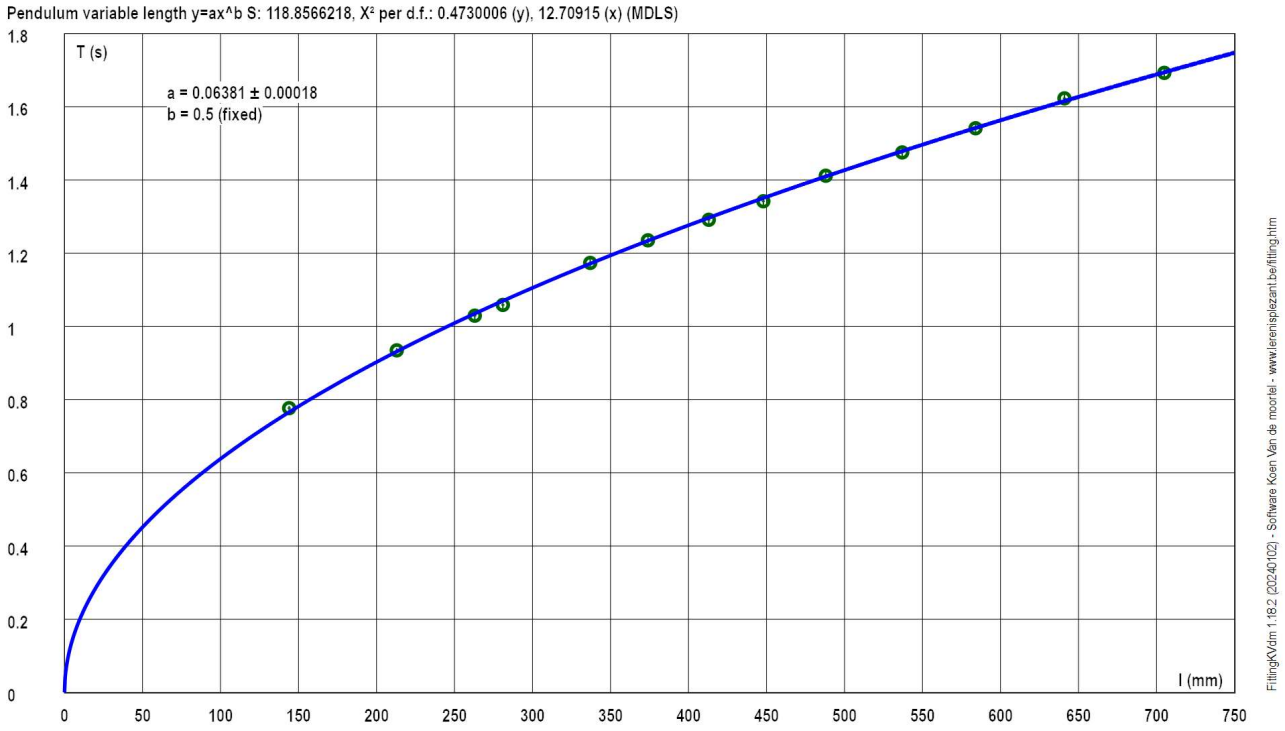
* Pendulum variable length.dta1

The bigger the length (l) of a pendulum, the bigger its period (T):

$$T \approx 2\pi\sqrt{\frac{l}{g}} = \frac{2\pi}{\sqrt{g}} \cdot \sqrt{l} = a \cdot l^b$$

with ideally: b = 0.5.
If you measure some periods, you can calculate g from the obtained best fitting a.

I did some measurements with a small metal object on a very thin rope.
Variables: x = length (in mm) from attachment point to the center of gravity of the object; y = average period (in s) of about 20 cycles.



From the fitted value of a, g can be calculated:

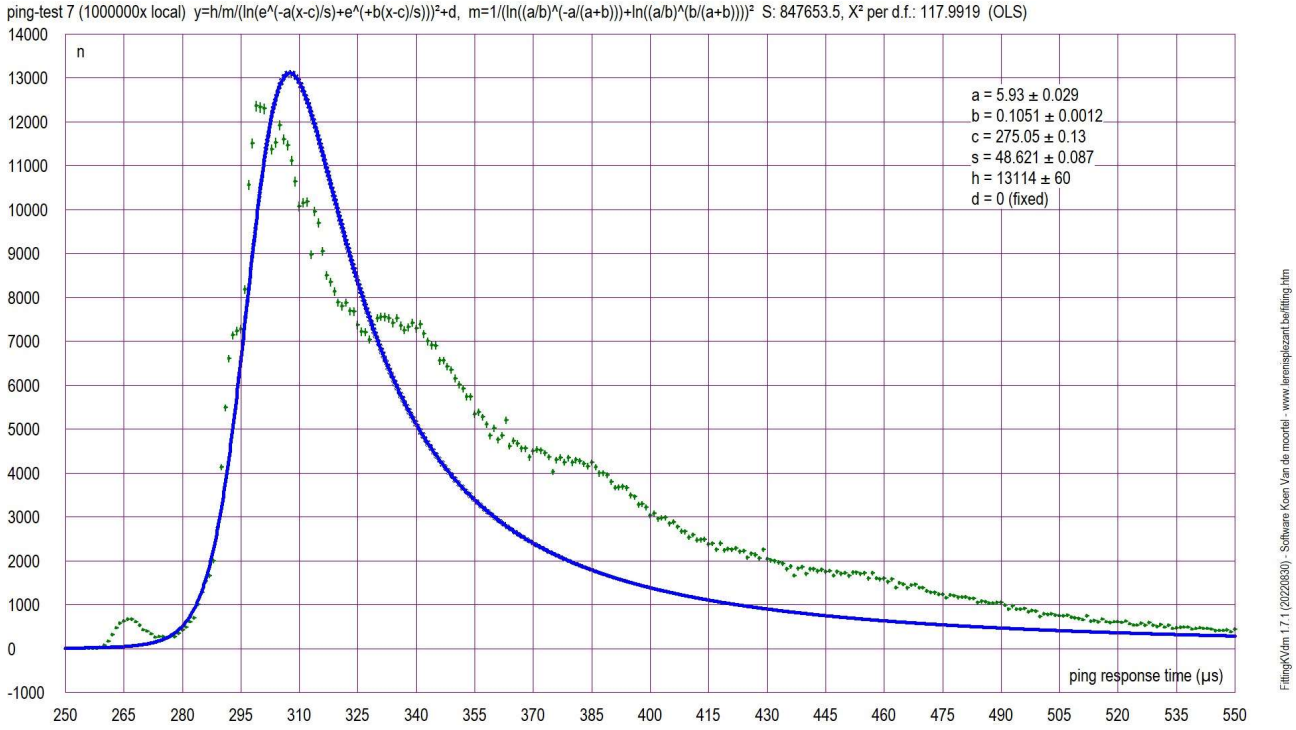
$$g = \left(\frac{2\pi}{a}\right)^2 \cdot \frac{1}{1000} \approx 9.69540 \frac{m}{s^2}$$ (/1000 because l was in mm)

This is a bit below the expected value on the location of the experiment (Gentbrugge, Belgium), but that may be because the construction was not sturdy enough.

* ping7.dta1

Variables: a million ping tests were done from my pc to a local IP address; x = the response time, rounded to 1µs, and y = the number of times this response time occured.

The result seems to a mix of different skewed distributions. You can try to fit the different peak functions but none of them describes the data perfectly of course. The "Skewed peak 1" fits reasonably well.

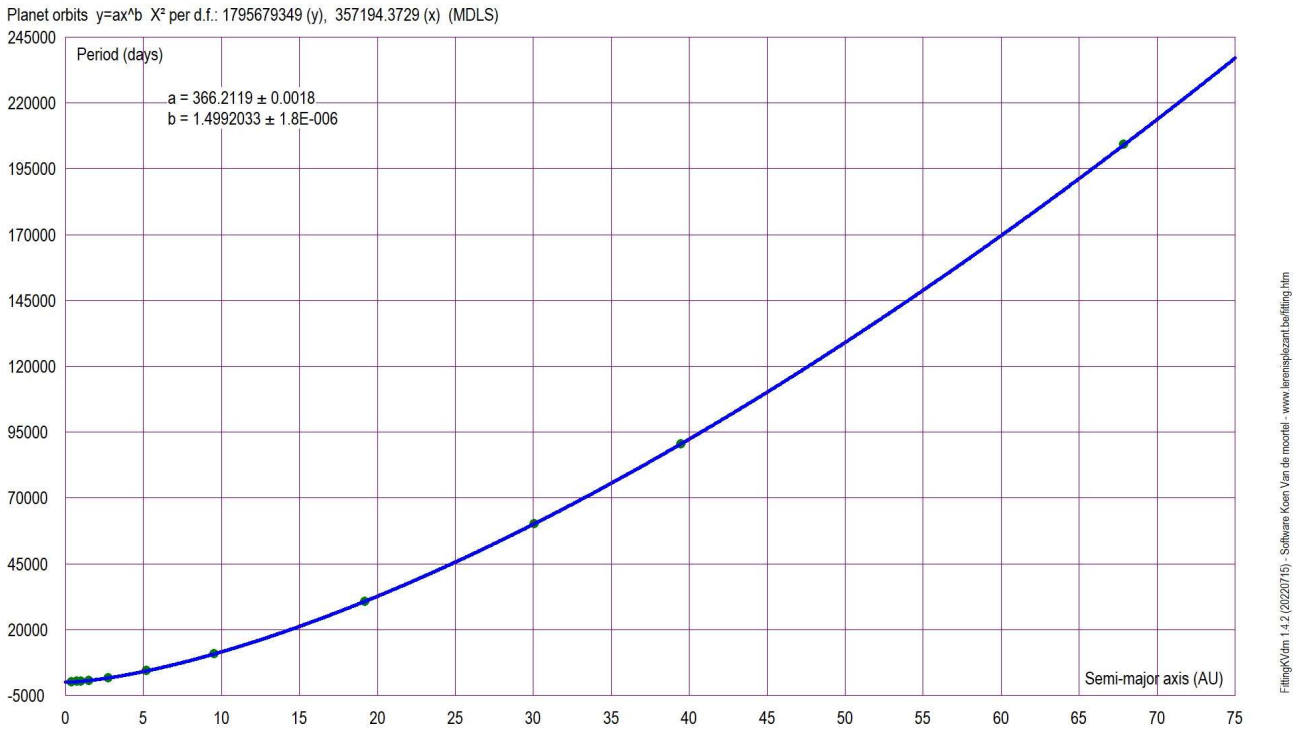


* Planet orbits.dta1

Variables: x = distance (semi-major axis) to the Sun (d, in AU = Astronomical Units); y = orbit period of the planet (T, in days). Kepler discovered that $T \sim d^{1.5}$ (his "third law").

Source: Wikipedia, who quotes from NASA obviously.

This law is pretty precise, as you can see:



Still, the high χ^2 values indicate that there are some other factors influencing the periods.

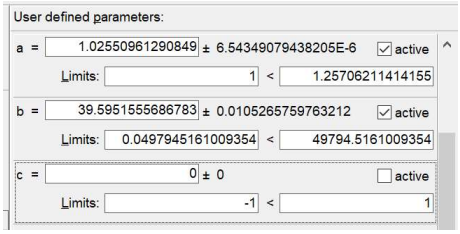
* **Population_Nigeria1950-2022.dta1**

Variables: x = time in years starting from 1950 (so x=60 means the year is 2010); y = population of Nigeria. The error on x was set to 0.5 since there was no information about the exact time of the year the population was counted.

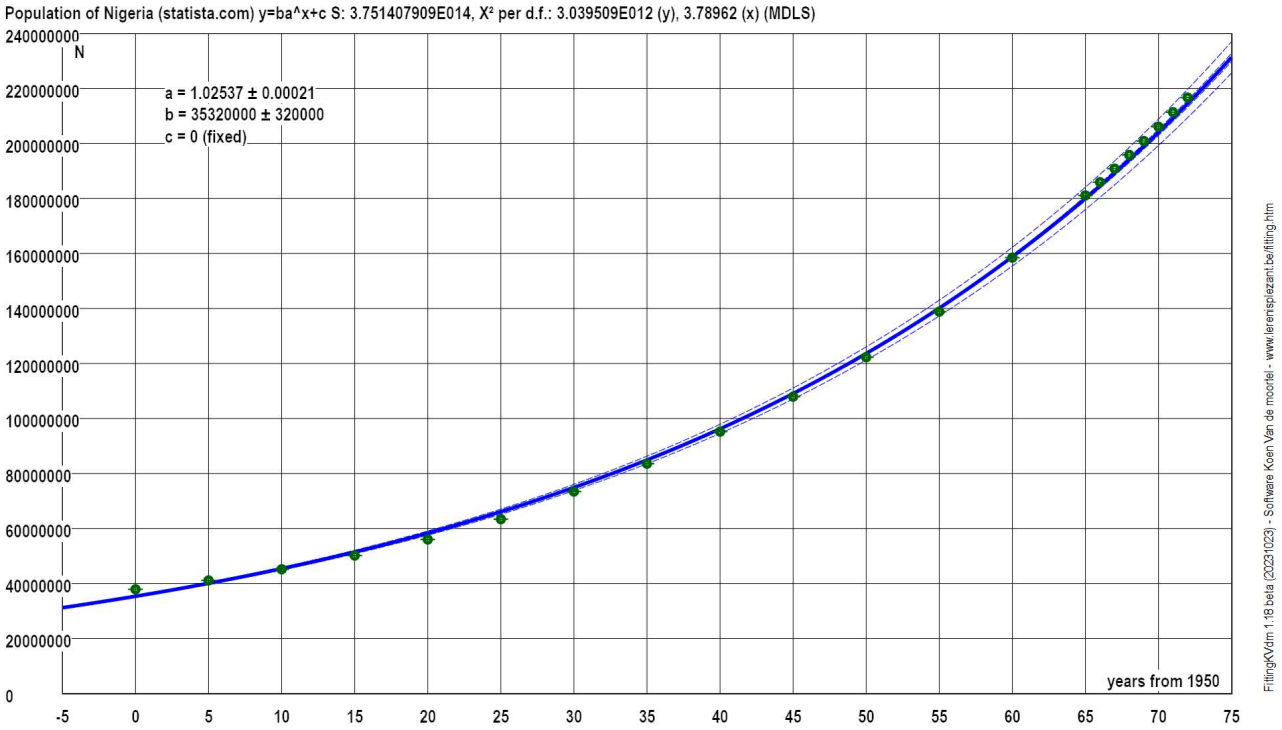
Source: www.statista.com/statistics/1122838/population-of-nigeria.

Usually, populations tend to show an exponential growth if there is enough food and space.

Try the "Exponential" model to check this. Block the baseline (c) to 0, since there is no reason to expect it to be different from 0.



You can see that the "Quadratic" and "Cubic" models also fit well, but their extrapolations make no sense.

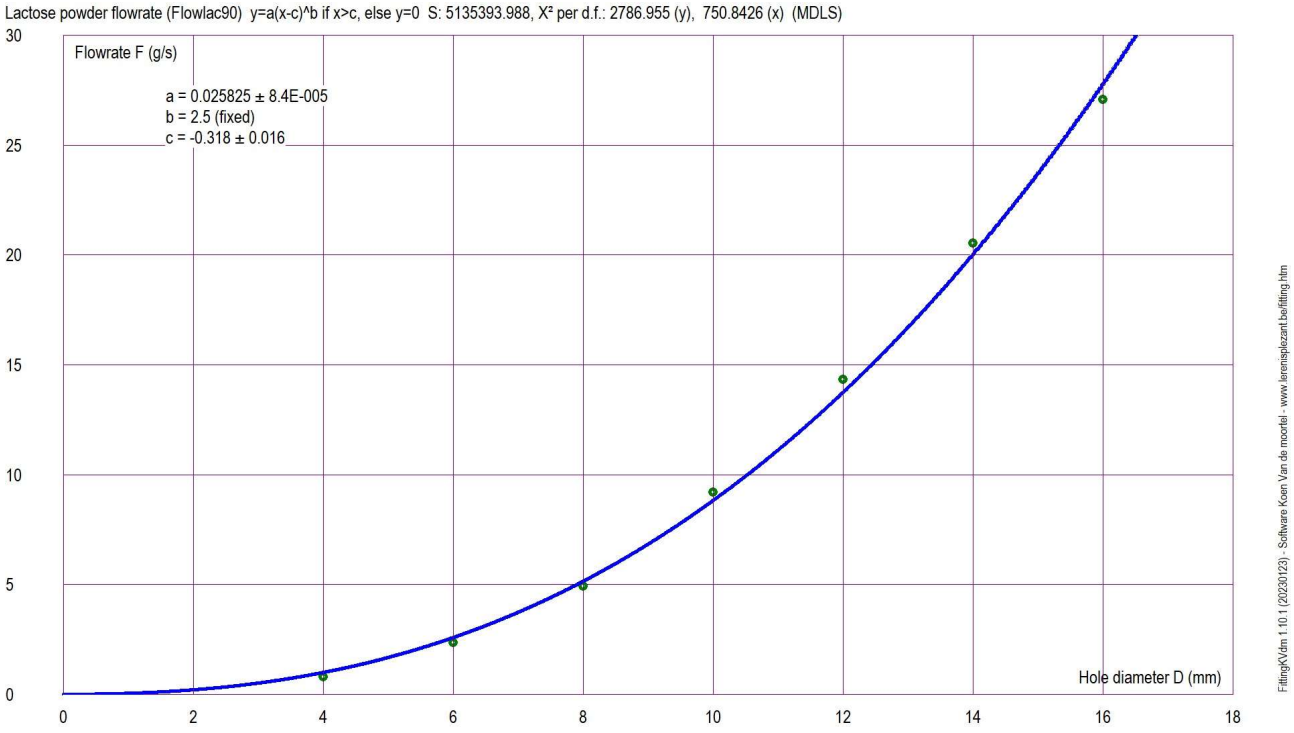


* Powder flowrate (Flowlac90).dta1

Variables: x = diameter of a hole were a powder was falling through (in mm); y = flow rate (in g/s). The powder was a kind of lactose, called "Flowlac90" from Meggle Pharma. The measurements were done with a device from Granutools.

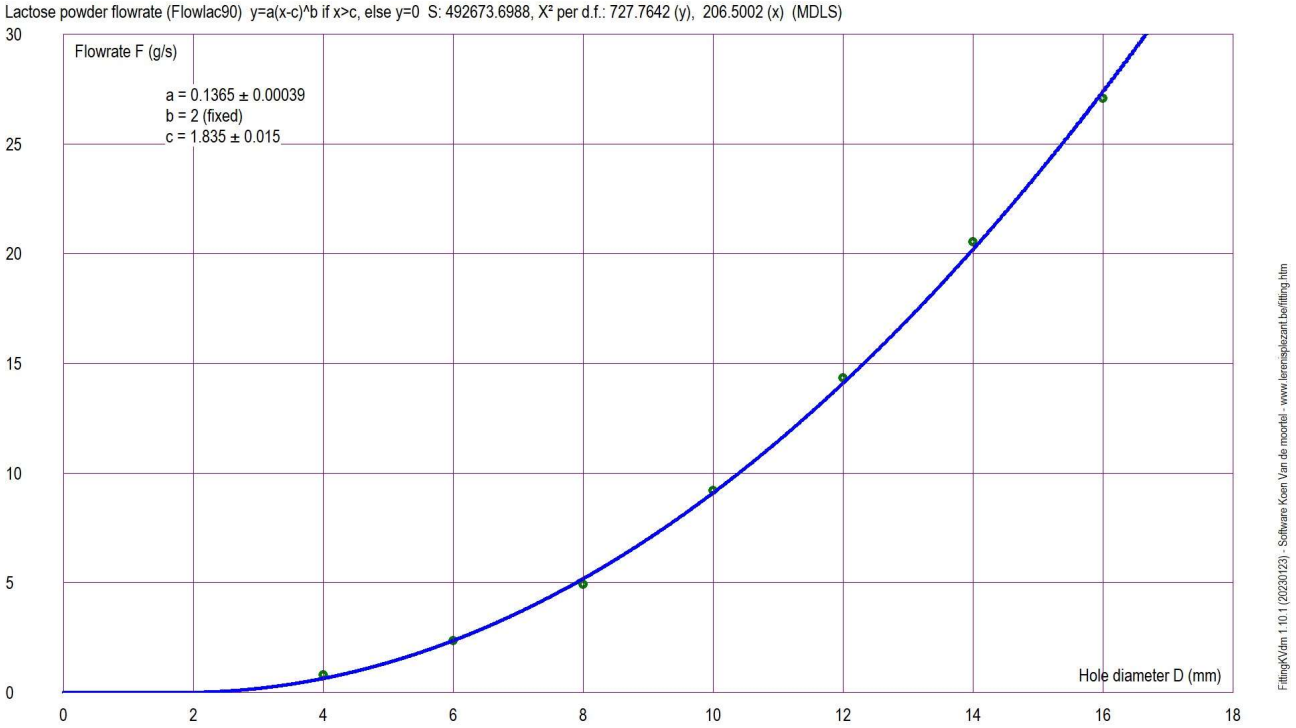
Source:
www.news-medical.net/whitepaper/20190910/A-Guide-to-the-Flowability-Classification-of-Lactose-Powders.aspx

In a reasonable approximation, the data fit with the "Power with horizontal shift" model with b = 2.5, according to the so-called Beverloo's law. The c parameter is supposed to be the size of the powder particle, but strangely, the fitted value is negative (with MDLS, and even more negative with OLS), not 0 as mentioned in the article.



The strange result might be explained by the simplistic model assumptions that the particles are round and without mutual attraction (cohesion).
Improved models have been suggested, e.g.:
www.unav.edu/documents/15083165/15313444/agm1_TGF07.pdf/f6b4d24e-b200-5dce-37b2-6495ca536370

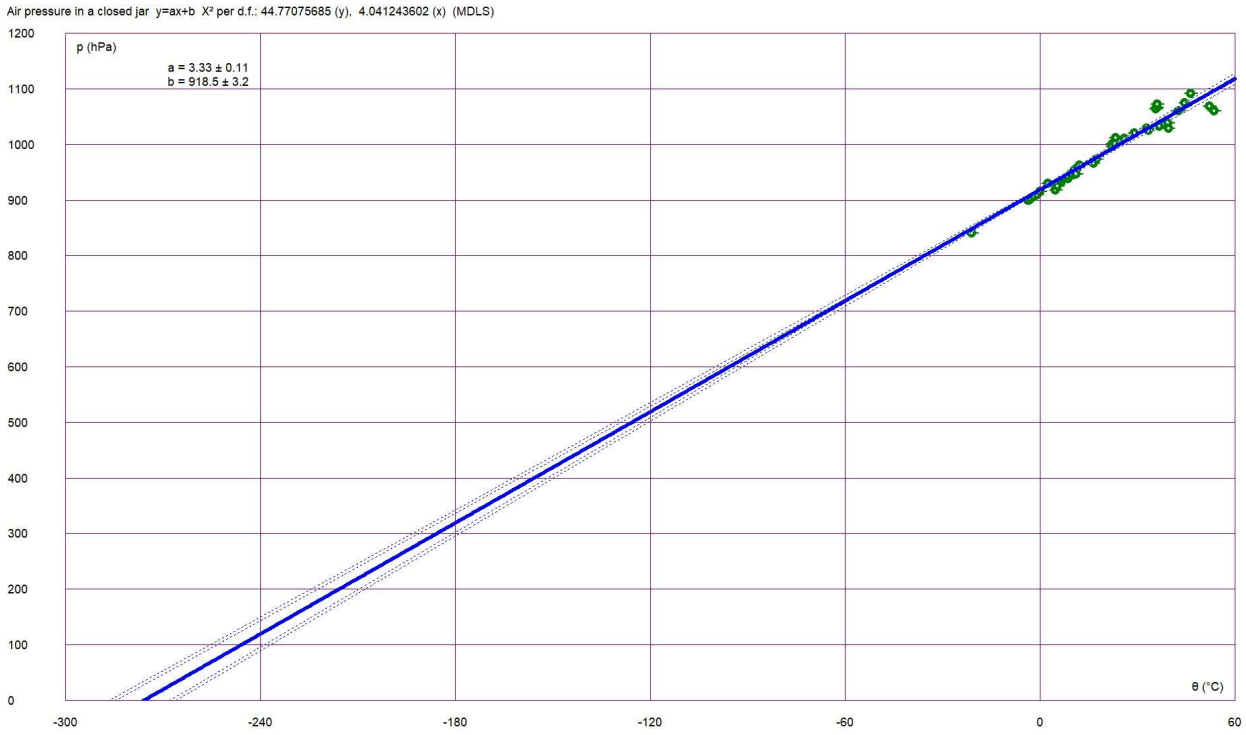
Or... and this makes more sense to me: the flow rate could just be proportional to the square of the diameter.
It fits ten times better, and c becomes positive:


Lactose powder flowrate (Flowlac90) y=a(x-c)^b if x>c, else y=0 S: 492673.6988, X² per d.f.: 727.7642 (y), 206.5002 (x) (MDLS)
Flowrate F (g/s)
a = 0.1365 ± 0.00039
b = 2 (fixed)
c = 1.835 ± 0.015
Hole diameter D (mm)
0
5
10
15
20
25
30
0
2
4
6
8
10
12
14
16
18
FittingKVdm 1.10.1 (20220123) - Software Koen Van de moortel - www.lerensplezant.be/fitting.htm

* Pressure_vs_temperature_in_jar.dta1

Variables: x = temperature (θ, in °C) in a closed jar; y = pressure of the air inside (p, in hPa).
The measurements were done with very simple home equipment: a multimeter with a thermocouple (wide range but not very precise) and a trekking barometer.

Use the "Linear" model and the ideal gas theory to find the value of the absolute zero temperature (0 Kelvin = …°C) as the intersect with the x axis (the "derived parameter" "zero" z=-b/a)! Definitely use the MDSL algorithm since p and T can both be dependent or independent variable.

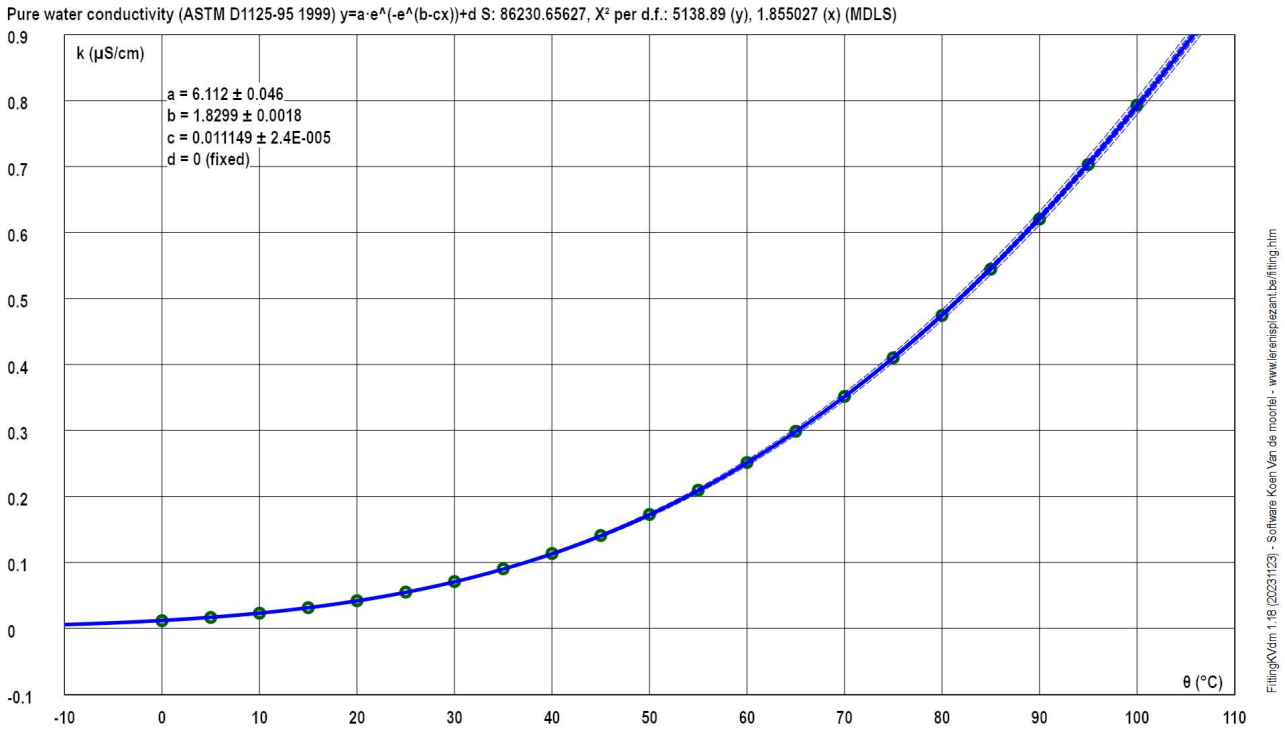


* Pure water conductivity vs T.dta1

Variables: x = temperature (θ, in °C); y = conductivity (k, in µS/cm) of pure water.

Source:
https://en.wikipedia.org/wiki/Conductivity_(electrolytic)#/media/File:Conductivity_of_Pure_Water.svg

The conductivity of water (pure or with dissolved products) increases with temperature, since the ions move faster. Even in pure water, there are some ions because of the dissociation of the water molecules.
The relationship of k vs θ (or absolute temperature T) looks a bit like exponential, but a Gompertz growth function fits much better:

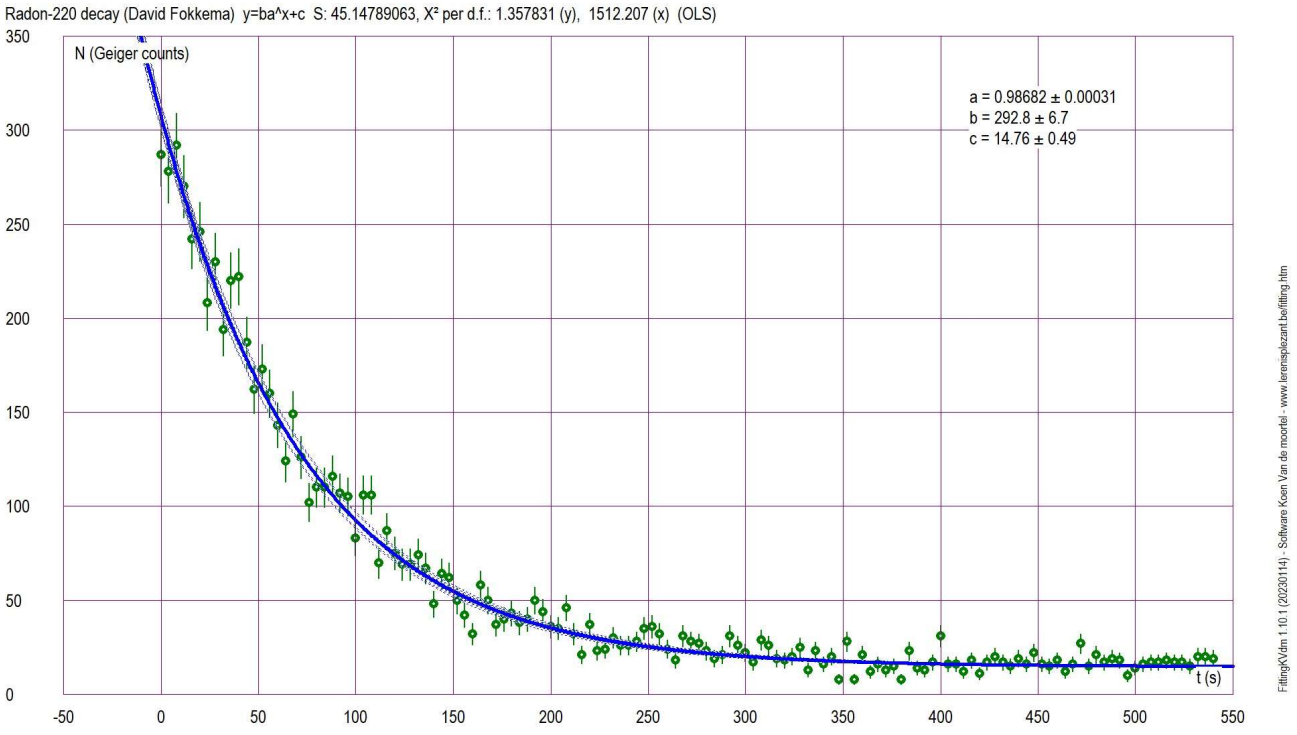


* Radon220 decay.dta1

Variables: x = t = time in seconds; y = N = counts from a Geiger counter near a short living radioactive source (Radon-220). σ_x = 2 because the count intervals were 4s. $\sigma_y = \sqrt{y}$ since it is a count.

Source: measurements done at the University of Amsterdam, kindly provided by Frank Fokkema.

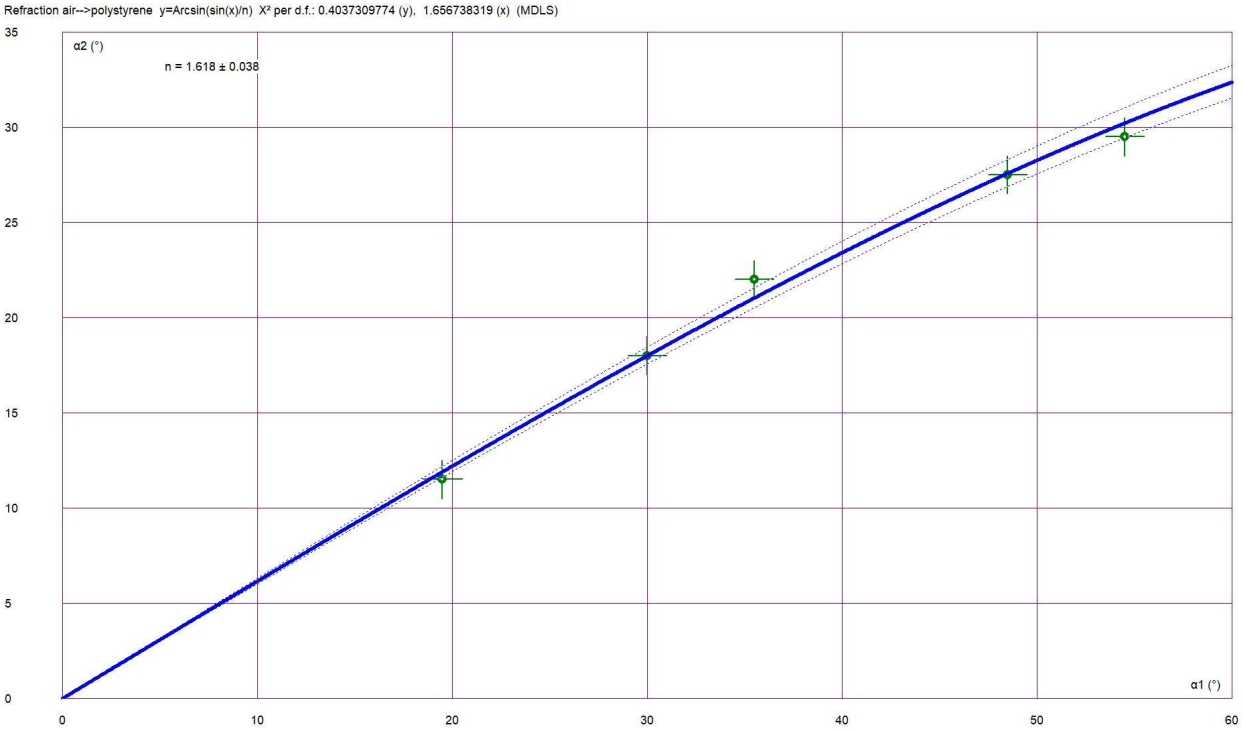
As expected, the "Exponential+baseline" fits well. Parameter c is the background radiation. The derived parameter "half-life" = 52.2 ± 1.2s. According to Wikipedia this should be 55.6s, but Brittanica says 51.3, see:
https://www.britannica.com/science/radon#ref237175
Since the data contain many points in the asymptote area, with a lot of noise, OLS is recommended.



* Refraction_polystyrene.dta1

Variables: x = incoming angle (α_1, in °, measured with a simple protractor); y = angle of the refracted laser beam (α_2, in °) going through a cd box (made of polystyrene).
The laser pointer had a wavelength of 532nm.

If you use the "refractive index" model and multidirectional fitting, you will get a good estimation of that index (should be 1.5983 according to www.refractiveindex.info).



*** Rivers.dta1**

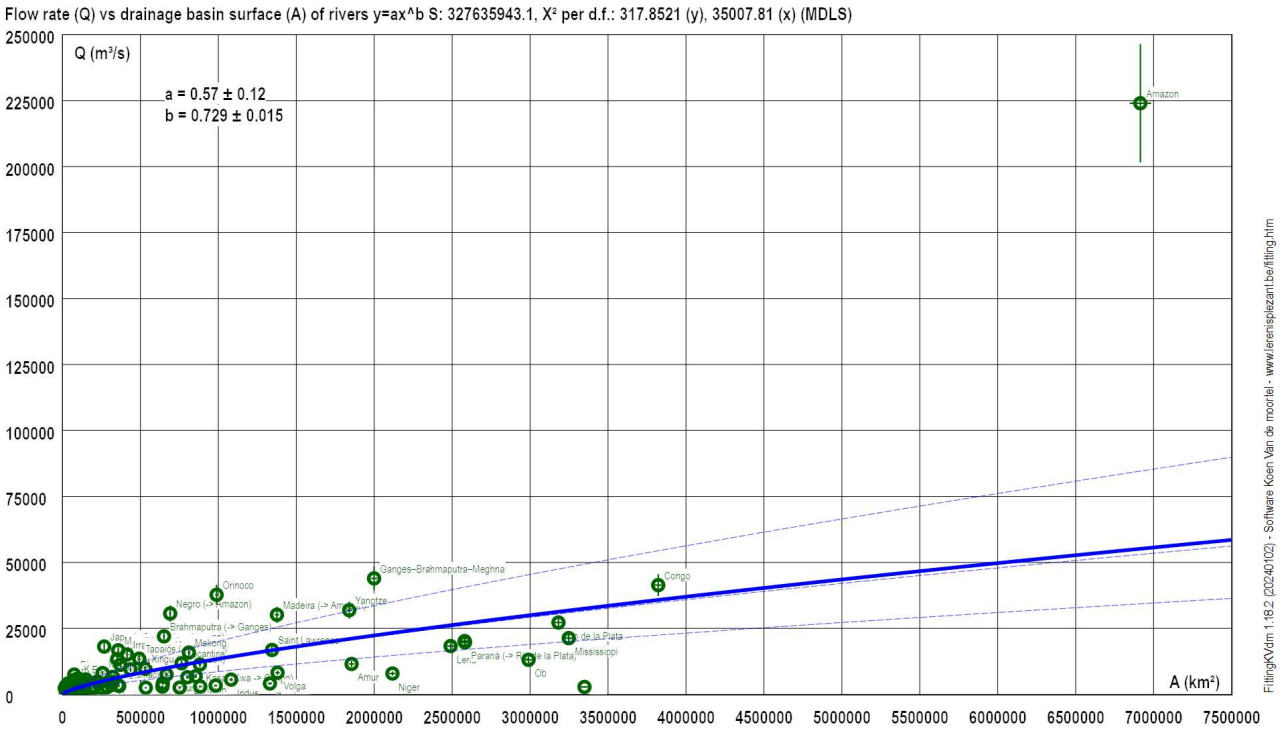
Variables: x = drainage basin surface (A, in km²) of a river; y = flow rate (Q, in m³/s).

Source: Wikipedia

A big surface collects more rain, causing a higher Q, but is there an exact relationship?

The the “Power” and the “Logarithm shifted through 0” both make sense!

You see, it doesn’t fit very well, because other variables like rainfall and temperature play a big role. The dot high above on the right, from the Amazon, shows that this river must have a lot of rainfall.



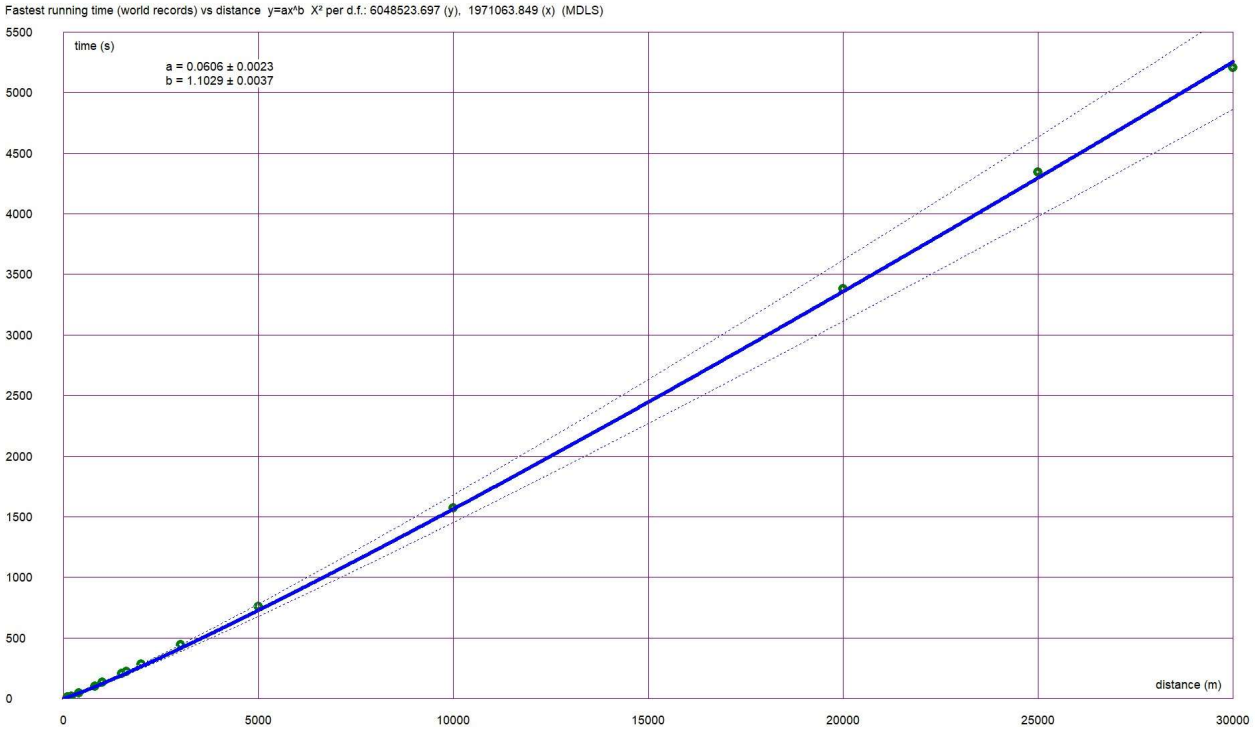
* Running records.dta1

Variables: x = distance in meters; y = fastest running time (world record) for that distance.

Source: Wikipedia.

At first sight, you might expect a linear relationship, since the time needed is proportional with the distance, but a runner is not a car or a train, a runner gets tired… So better try the "Power" model. You will find that the exponent is not 1, but a bit higher.

Notice the big χ^2 values, even though the curve fits very nicely. Why are they so big? Well, the given σ_x and σ_y values are the precisions of the distance and time measurements here. χ^2 would be much lower if the y and σ_y values would be averages and standard deviations of a group of random runners.



* **Running speeds ultra-marathons men.dta1**
* **Running speeds ultra-marathons women.dta1**

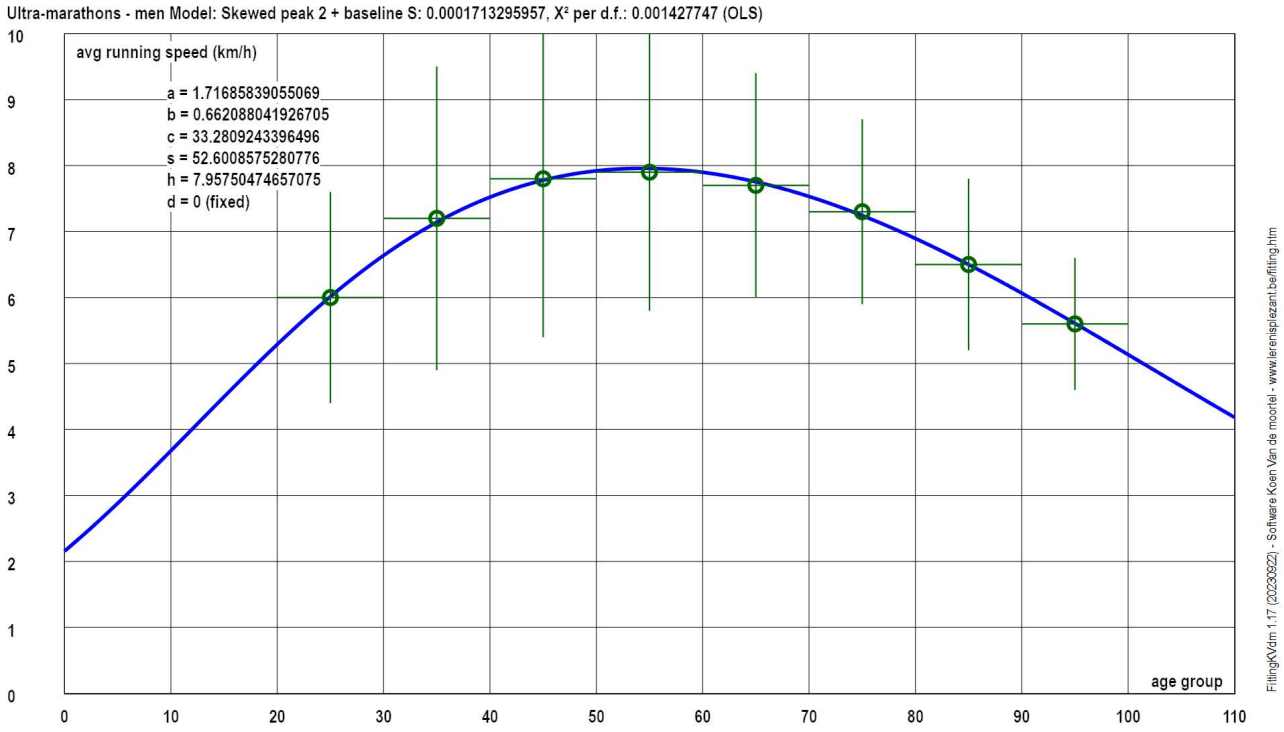
Variables: x = center of an age group in years (e.g. for ages [40, 50[: x = 45±5), y = average running speeds for 100km ultra-marathons ± SD in km/h.

Source:
Angelika Stöhr et. al.: "An Analysis of Participation and Performance of 2067 100-km Ultra-Marathons Worldwide", Int. J. Environ. Res. Public Health 2021, 18(2), 362; https://doi.org/10.3390/ijerph18020362 or https://www.mdpi.com/1660-4601/18/2/362

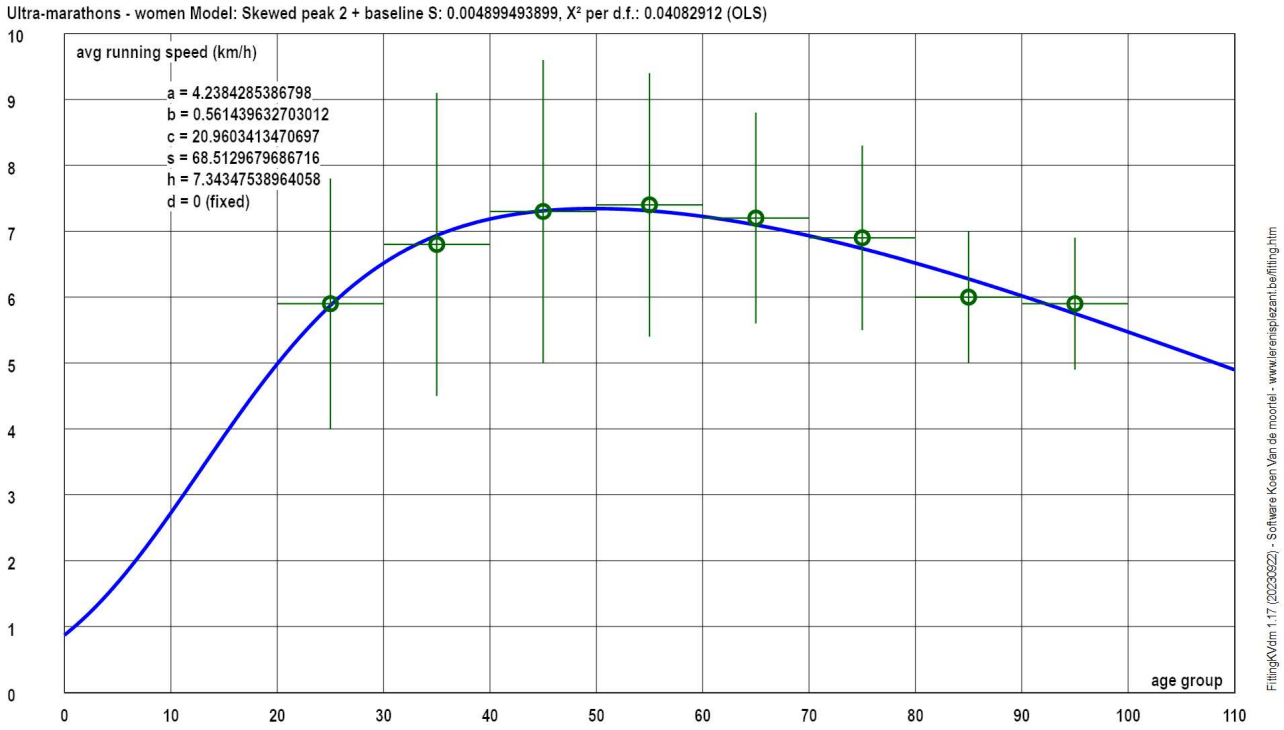
I had no idea that so many 90+ people could still run 100km, but apparently they do, and not even that much slower than younger people.
The "Skewed peak 2" model fits nicely, with d = 0 (we suppose that the speed must come to zero at a certain age).

Warning: there are 5 parameters involved and only 8 data points, so the risk of overfitting is real here. I wouldn't trust any extrapolation of the curve, neither left or right!
I expect the fit would be more realistic with the raw data, not grouped.

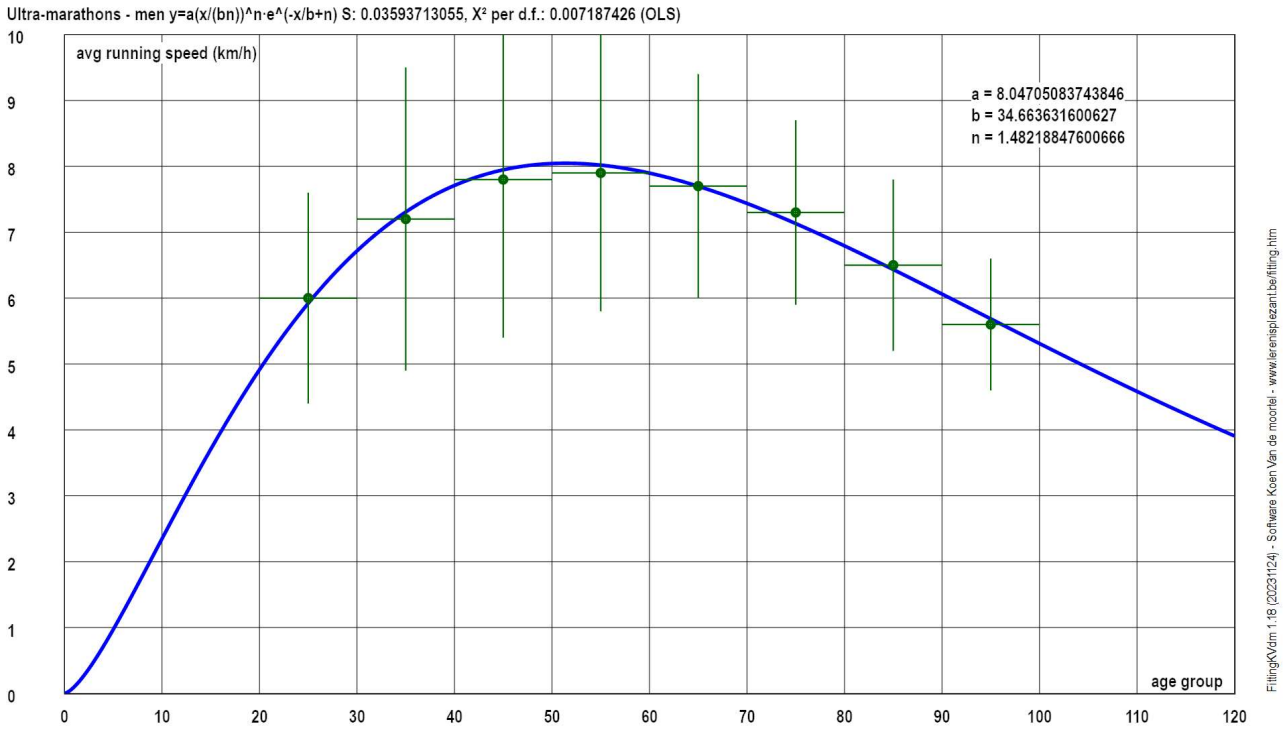
These are the results for men:



and for women:



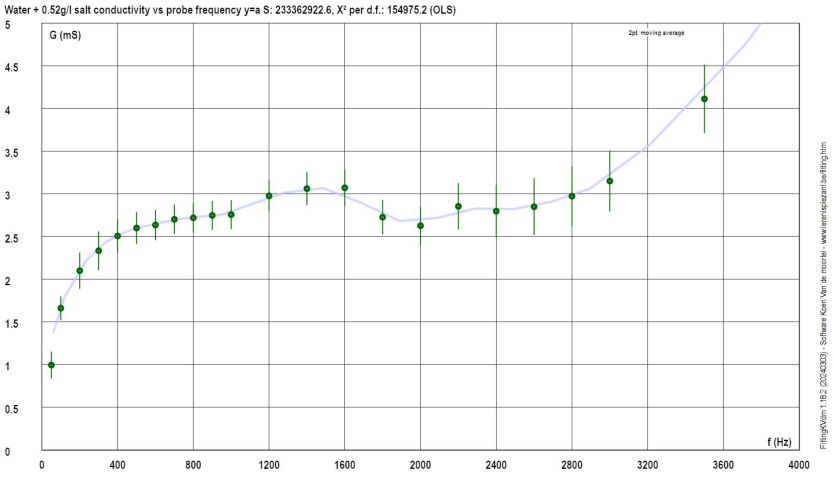
A model that produces a bit more realistic behavior for early ages, is “Power•Exp.decay”:



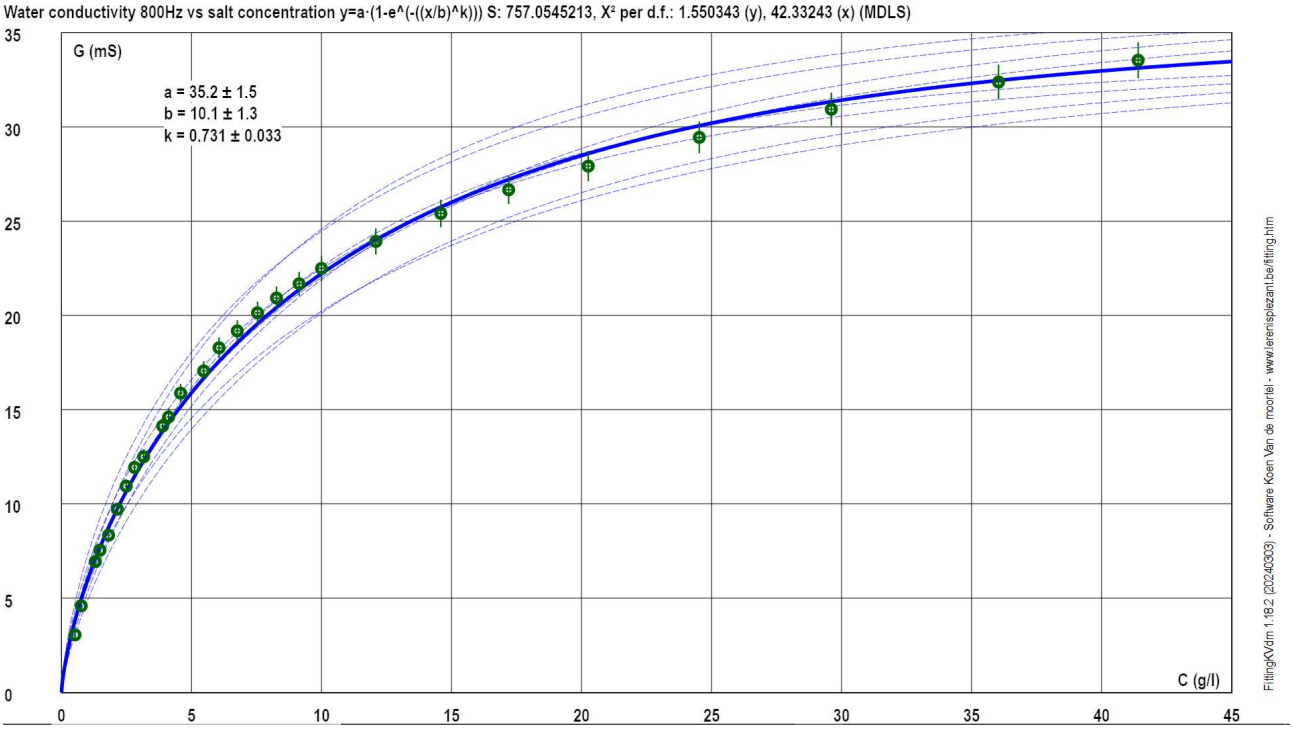
* **Salt_water_G_vs_f.dta1**
* **Salt_water_conductivity_800Hz.dta1**

Variables: y = conductivity of water with salt, measured with a simple probe (a recycled European 220V plug); in the first file: x = frequency produced with laptop, taken from the audio output, with a salinity of 0.52g/l; in the second dataset, the frequency was kept constant (800Hz) and x = salinity in g/l. The measurements were done by myself.

The first one shows a strange pattern; the 2pt. moving average looks like this:



The second one shows a nice pattern that can be approximated by a Weibull growth function, precisely enough to be used as a calibration for the probe. You can convert a measured conductivity to a salinity with it.



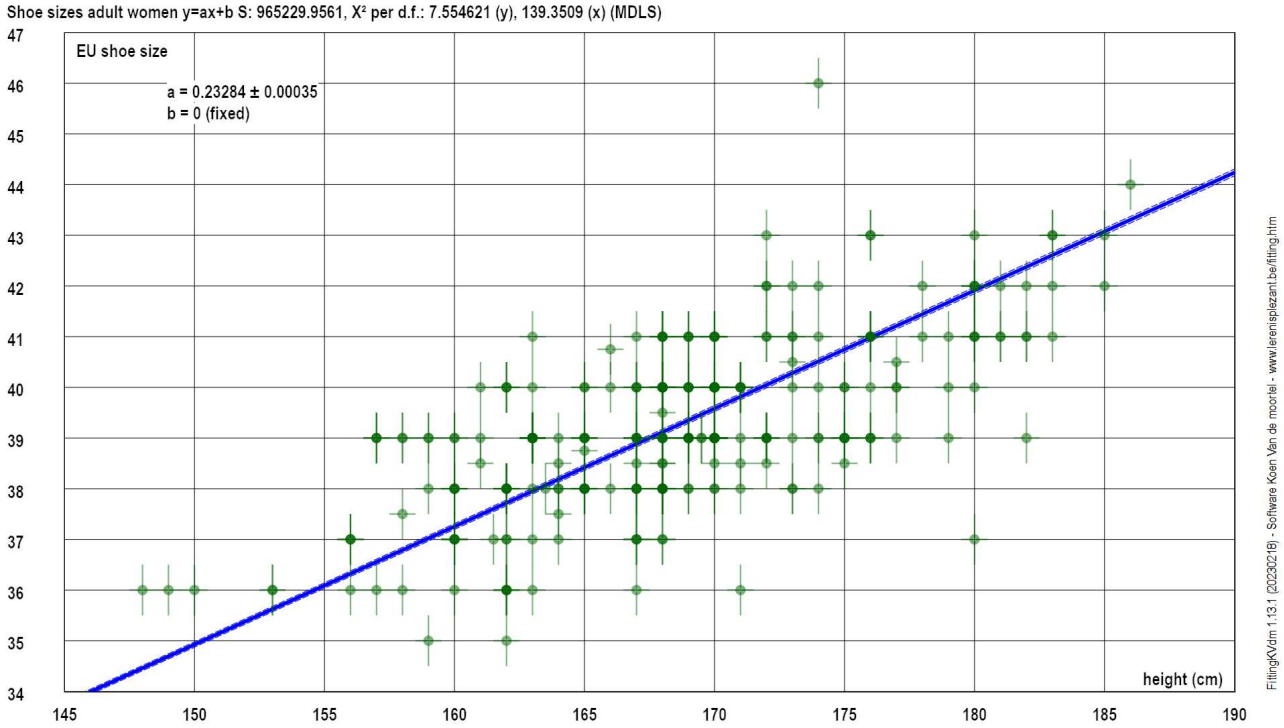
Also try the logarithmic function shifted through the origin; it fits almost as good.

* **Shoe sizes adult men.dta1**
* **Shoe sizes adult women.dta1**

Variables: x = height in cm of a person, y = (European) shoe size.
Since the heights were rounded to 1cm, the "error" on x was set to 0.5. The shoe sizes are mostly whole numbers, so someone's "best fitting" shoe size could also be 0.5 higher or lower.

Source: The data were collected by myself from mainly Belgian and Dutch people on social media in Feb. 2023.

A linear model with b=0 (since 0 height = 0 shoe size) fits well. Here are the women:

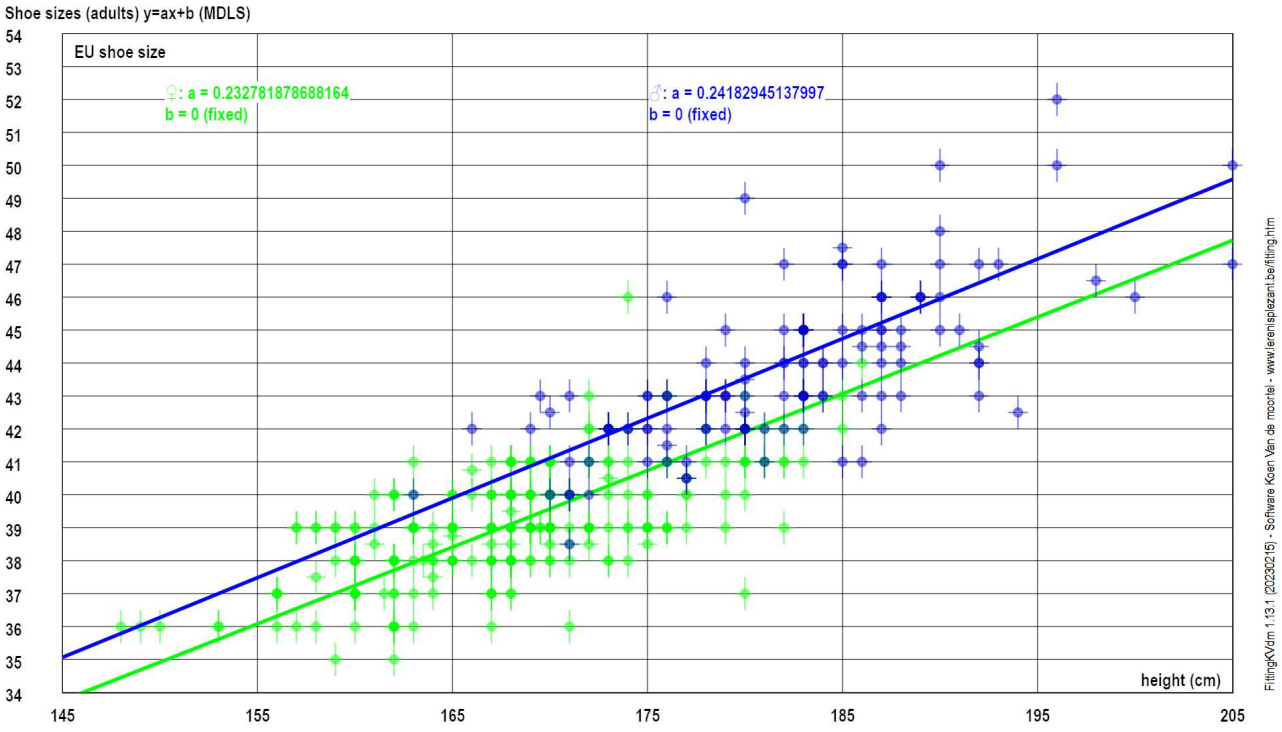


The graph option "transparent points" was chosen, so you can see which measurements occured multiple times (darker points).

The "power" model is worth trying too, since it might be that feet lengths and heights are not perfectly proportional! You will definitely notice a difference between OLS and MDLS then!

Remark

If you save the graphs of the men and the women with the same scales, you can open the SVG files with a plain text editor like Windows Notepad, change the color settings in one file, and then copy and paste the code parts where the dots and lines are drawn into the other file, you can obtain a composite graph like this:

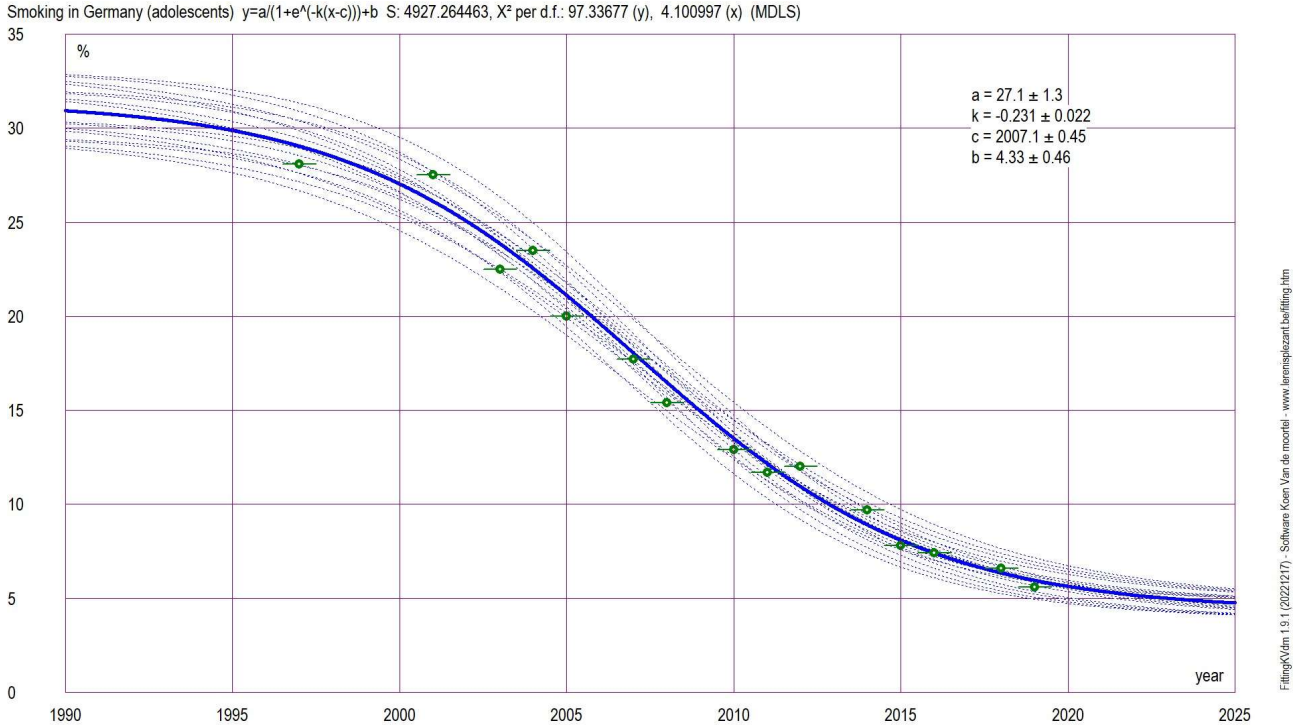

Shoe sizes (adults) y=ax+b (MDLS)
EU shoe size
♀: a = 0.232781878688164
b = 0 (fixed)
♂: a = 0.24182945137997
b = 0 (fixed)
height (cm)
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
145
155
165
175
185
195
205
FittingKVdm 1.13.1 (20230215) - Software Koen Van de moortel - www.lerenisplezant.be/fitting.htm

* Smoking - adolescents Germany.dta1

Variables: x = time (year); y = percentage of adolescents smoking. Since we didn't know at what time in the year the survey was done, or when people started or stopped smoking, σ_x was set to 0.5, so x can be interpreted as "the middle of the year ±0.5 year. Looking at the percentages, we can guess σ_y to be 0.05, or safer 0.1. Anyway if all the errors are the same, the absolute value will not influence the parameters, only the parameter confidence interval estimations.

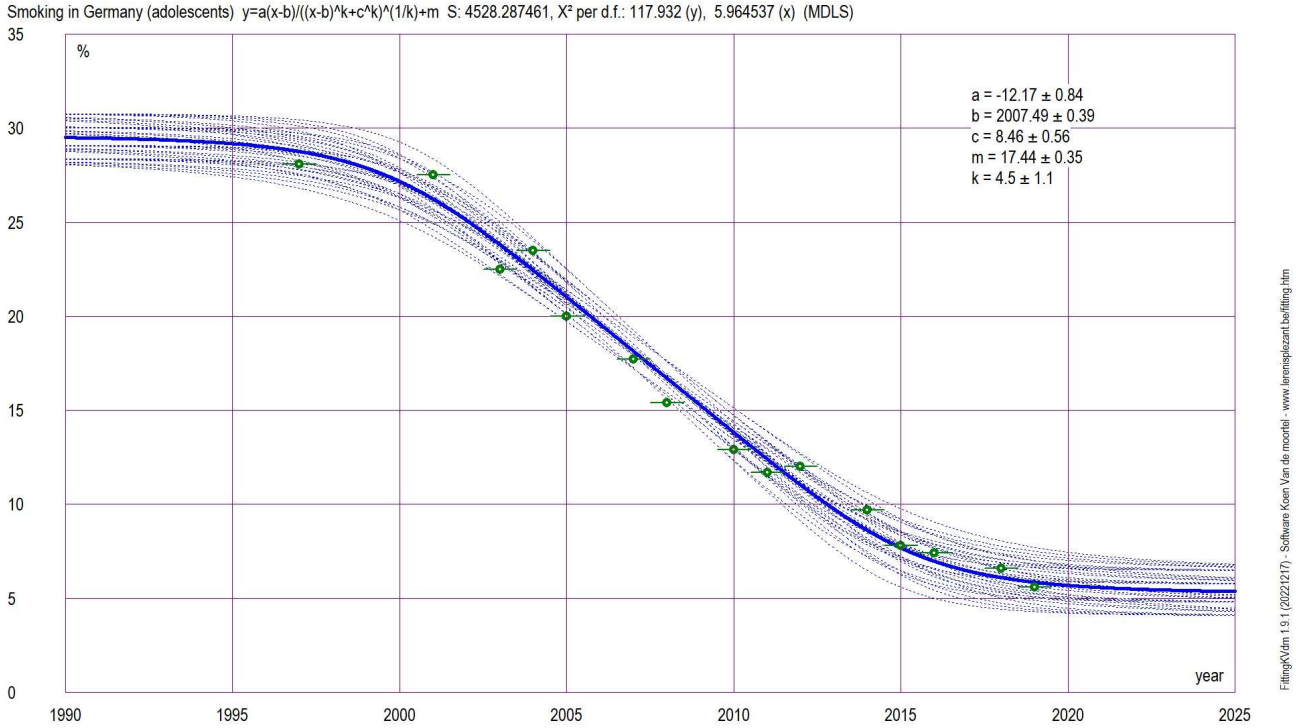
Source:
https://www.destatis.de/EN/Themes/Society-Environment/Sustainable-Development-Indicators/Publications/Downloads/data-relating-indicator-report-2021.pdf?__blob=publicationFile

This is a typical example that might be modeled with a logistic function: a change in behaviour caused by an external stimulus (government campaigns against smoking).



The "Logistic with baseline" model (with MDLS) predicts that the percentage will eventually go from 27.1 to 4.33.

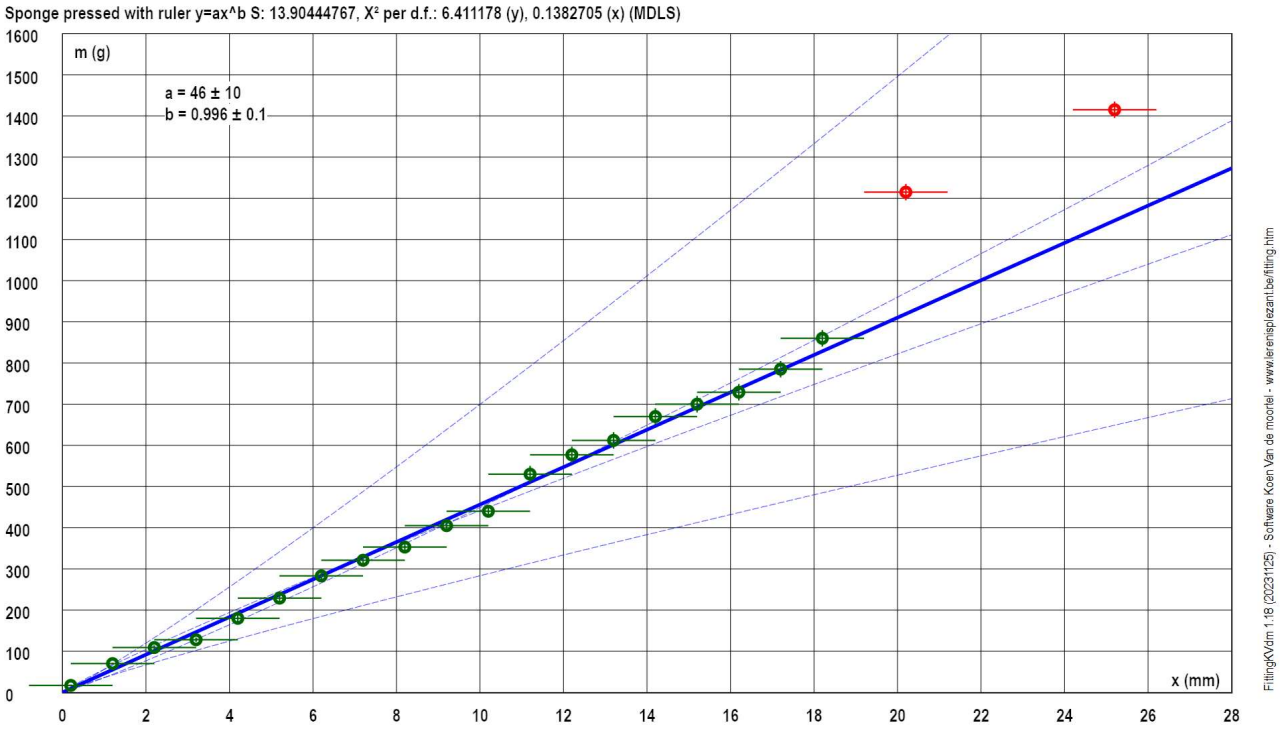
You might also try the "Transition" model. That predicts a percentage going from 29.6 to 5.27.


Smoking in Germany (adolescents) y=a(x-b)/((x-b)^k+c^k)^(1/k)+m S: 4528.287461, X² per d.f.: 117.932 (y), 5.964537 (x) (MDLS)
%
a = -12.17 ± 0.84
b = 2007.49 ± 0.39
c = 8.46 ± 0.56
m = 17.44 ± 0.35
k = 4.5 ± 1.1
year
35
30
25
20
15
10
5
0
1990
1995
2000
2005
2010
2015
2020
2025
FittingKVdm 1.9.1 (20221217) - Software Koen Van de moortel - www.lerensplezant.be/fitting.htm

* Sponge-ruler.dta1

Variables: a rectangular dry sponge was placed on a weighing scale, and then a ruler was pushed in slowly step by step vertically, in the middle of the top side of the sponge (height: 90mm). Variable x = distance (in mm) that the ruler was pushed in; y = m = the mass given by the scale (in grams). So the force F in Newtons was given by y/1000·9.81.
The last measurements (with m > 1kg) were unreliable since the sponge started to show plastic (inelastic) deformations.
Although the resolution of the scale was 1g, I estimate the error on the mass measurements to be about ±10g, since it was very difficult to keep the ruler totally stable, and hence the reading fluctuated a bit. A mechanical construction would improve the precision here.

Fitting the data with a power function, produced an exponent very close to 1, which suggests that the sponge obeyed Hooke's law, at least if you don't press it too hard.



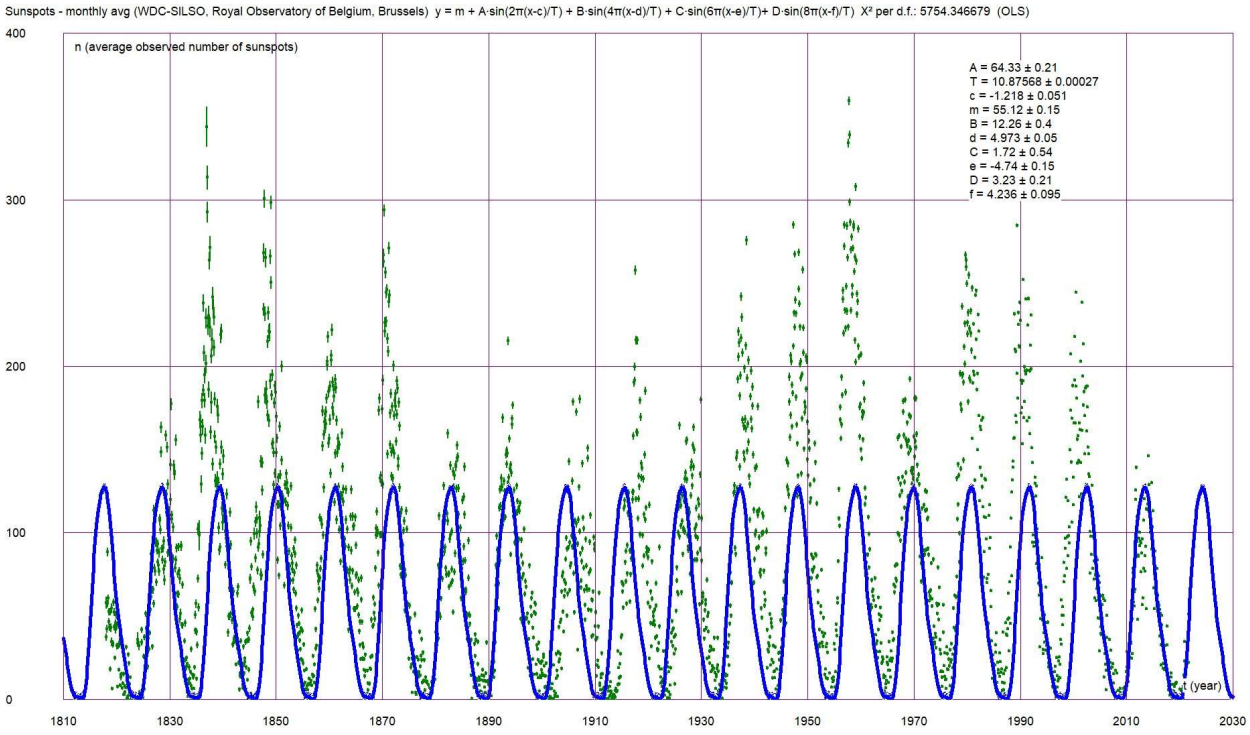
* Sunspots-monthly avg.dta1

Variables: x = time in years (1818-2021); y = monthly average of the daily counted sunspot number (& standard deviation of that average).

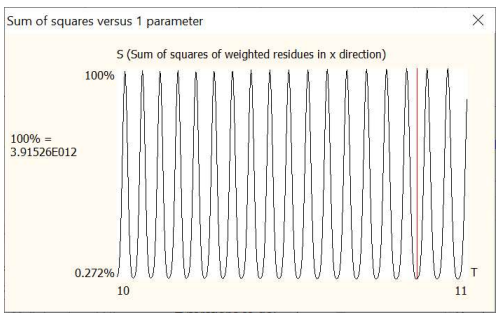
Source: WDC-SILSO, Royal Observatory of Belgium, Brussels,
See: www.sidc.be/silso/infosnytot

This is a tricky one! You clearly see a periodic pattern, but it is not very constant.
The "Sine wave with harmonics" model will fit reasonably, but only if you start with a good estimation of the period: T≈10.9 years (limits 10.7 and 11.1).
For the amplitudes of the main wave and the first harmonics A, B, C & D, you can enter 1 and limits 0 and 100; for the phase shifts c, d, e, f you can start with 0 and limits -6 and 6.
It will take a few thousand iterations to get a stable fit, and a long calculation time for the precision estimation of the parameters.

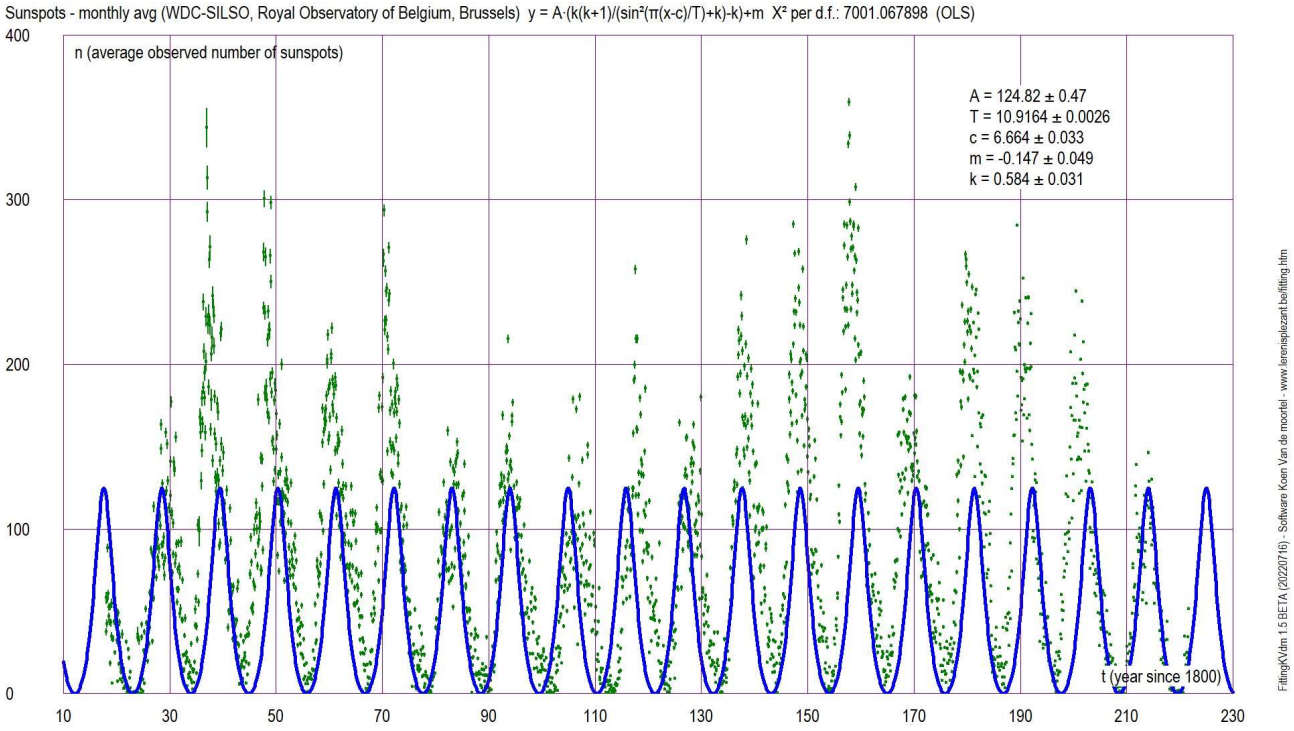
This is a good example to show the advantage of subtracting the biggest part of the time. It the original file, the time was given in years AD. In that case, a very small deviation will give an enormous difference after 1800 years, so you get convergence for many other starting values for T (even 10.8 or 11.0). With 10.9 it converges to 10.87568 with the lowest χ^2 value, but that's hard to find out.



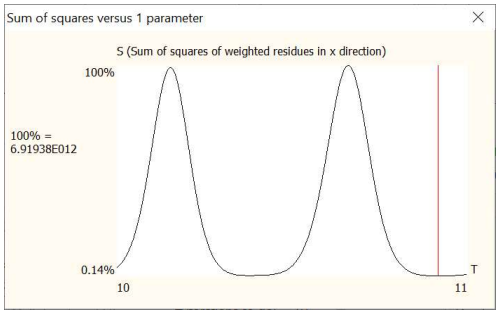
You can see this if you click on the parameter T panel:
S goes wildly up and down around T = 10..11.



You may also try the “Periodic peaks” model:



As you can see in the S vs T graph, the number of peaks (the minima are possible T values) has gone down dramatically because 1800 was subtracted from all the years.



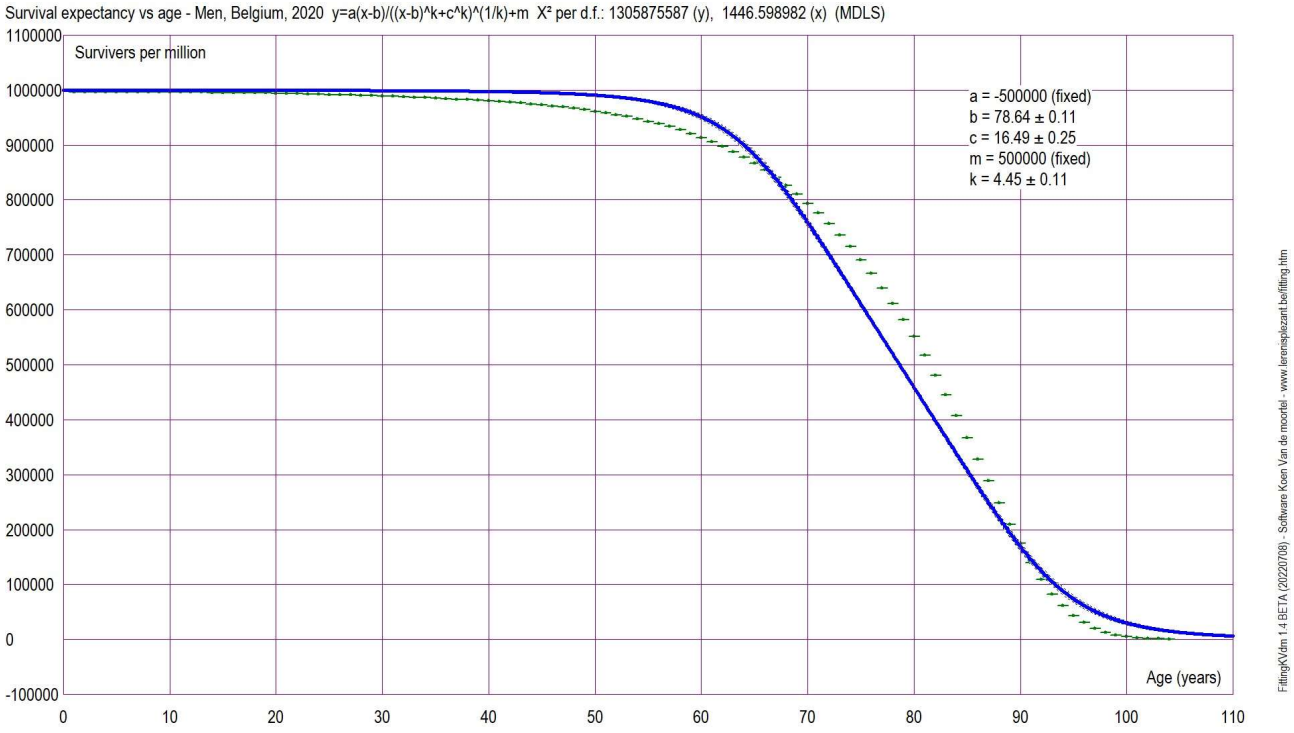
Since the periods and amplitudes clearly differ from time to time, the model is absolutely not perfect, but it might give reasonable predictions.

* Survival chances Men Belgium 2020.dta1

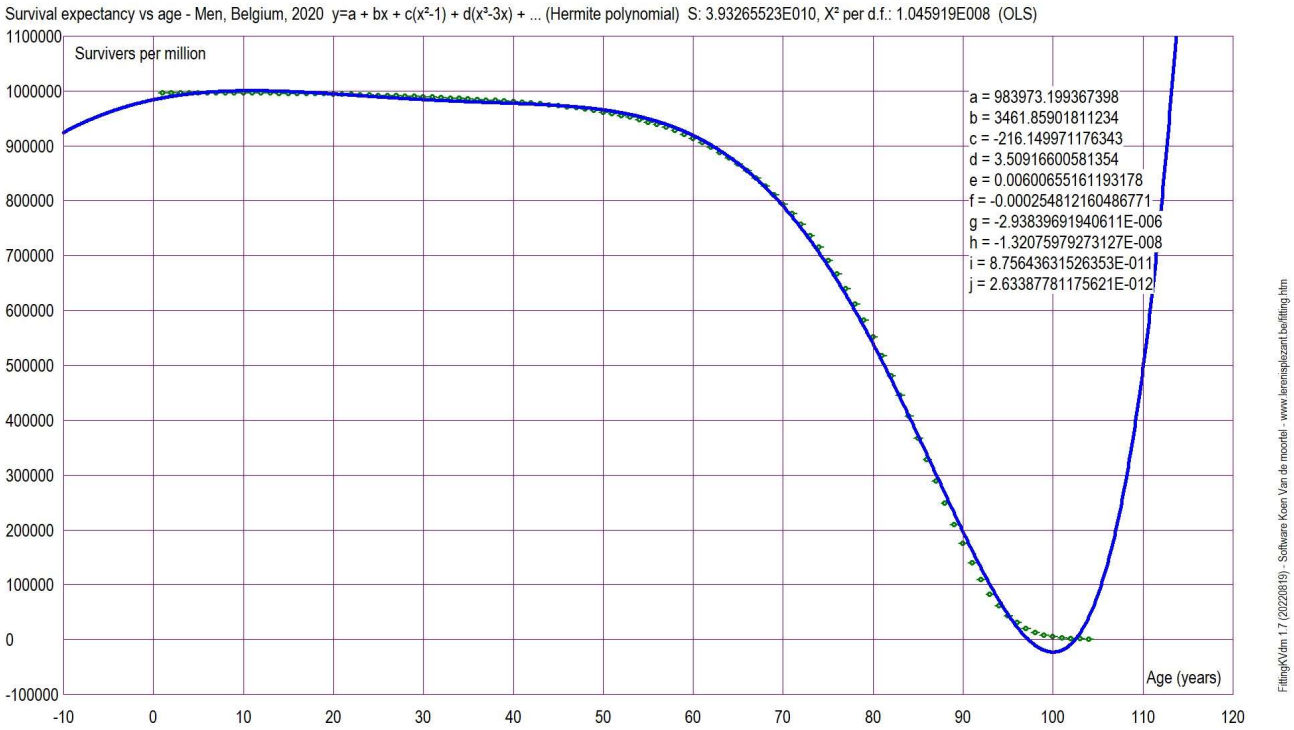
Variables: x = age (years), y = the average number of men that will survive until that age (Belgium, 2020). No standard deviations were available, so the "errors" on x and y were simply set to 0.5 and 1, which will cause the χ^2 values to be unrealistically high, but the influence on the parameters is most probably very low.

Source:
statbel.fgov.be/nl/themas/bevolking/sterfte-en-levensverwachting/sterftetafels-en-levensverwachting#figures

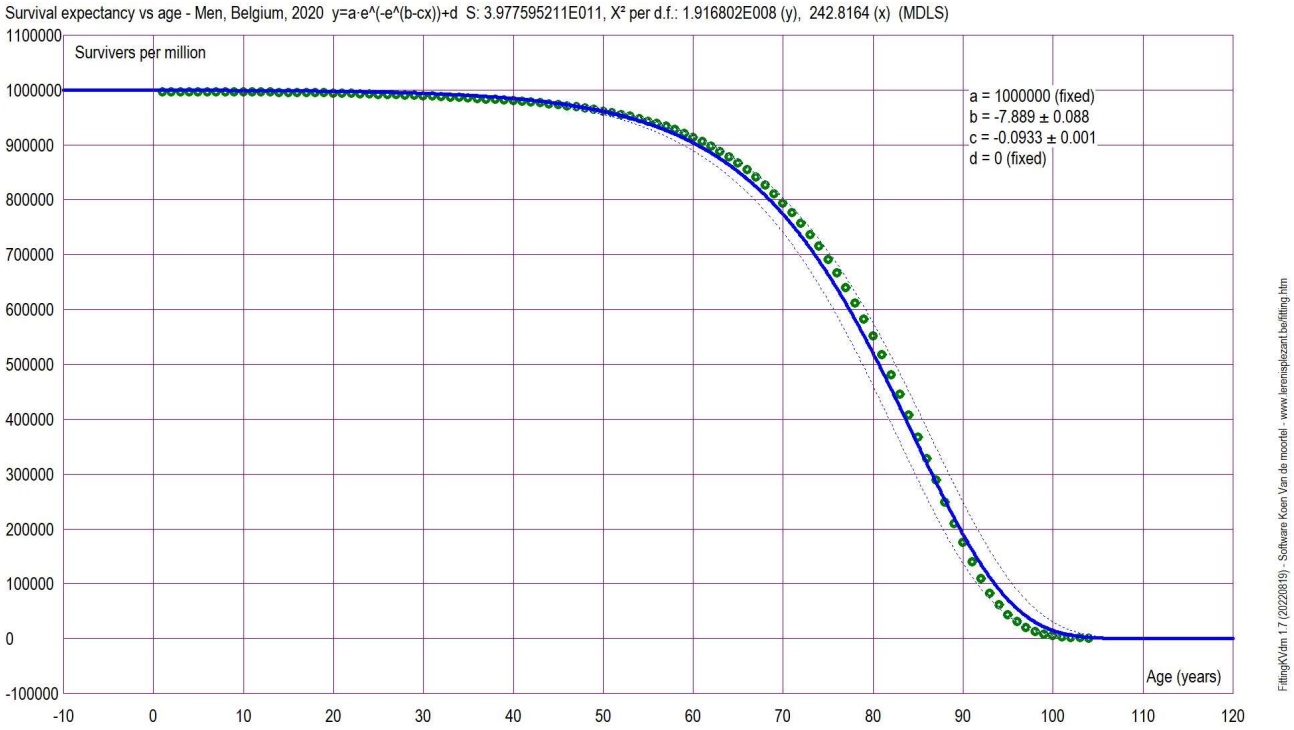
Requirements for the model: y should go from 1000000 to 0, so the "Logistic" (with fixed a = 1000000 and b = 0) and "Transition" (with fixed central value m = 500000 and "amplitude" a = 500000) are good candidates. Unfortunately, none of them really follows the pattern.



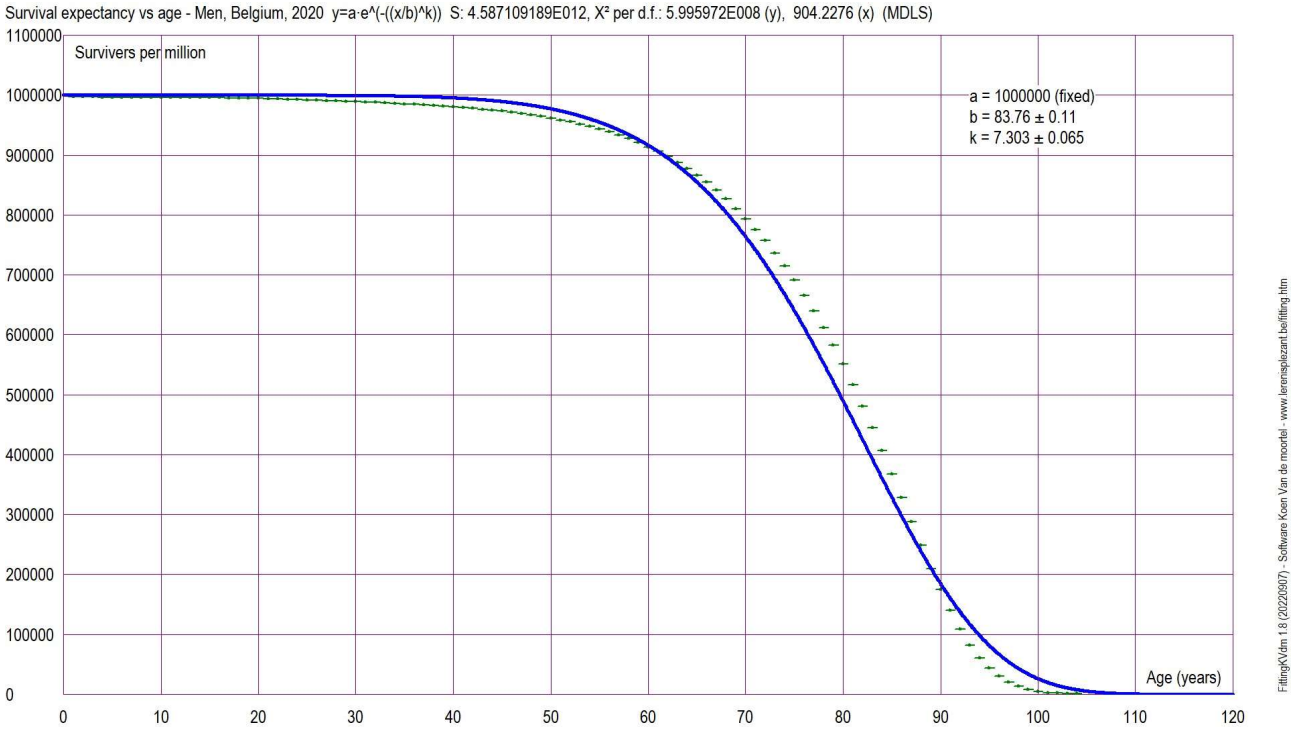
You might try the Hermite polynomial model here, and discover that the overall fit is "better", which might have some practical value to "predict" the expectancy value for some age, but the moment you look at the right or the left side of the graph, you see how ridiculous this model is.



Probably the best model here is the "Gompertz growth". Make sure to set the upper and lower limits fixed (a=1000000 and d=0):



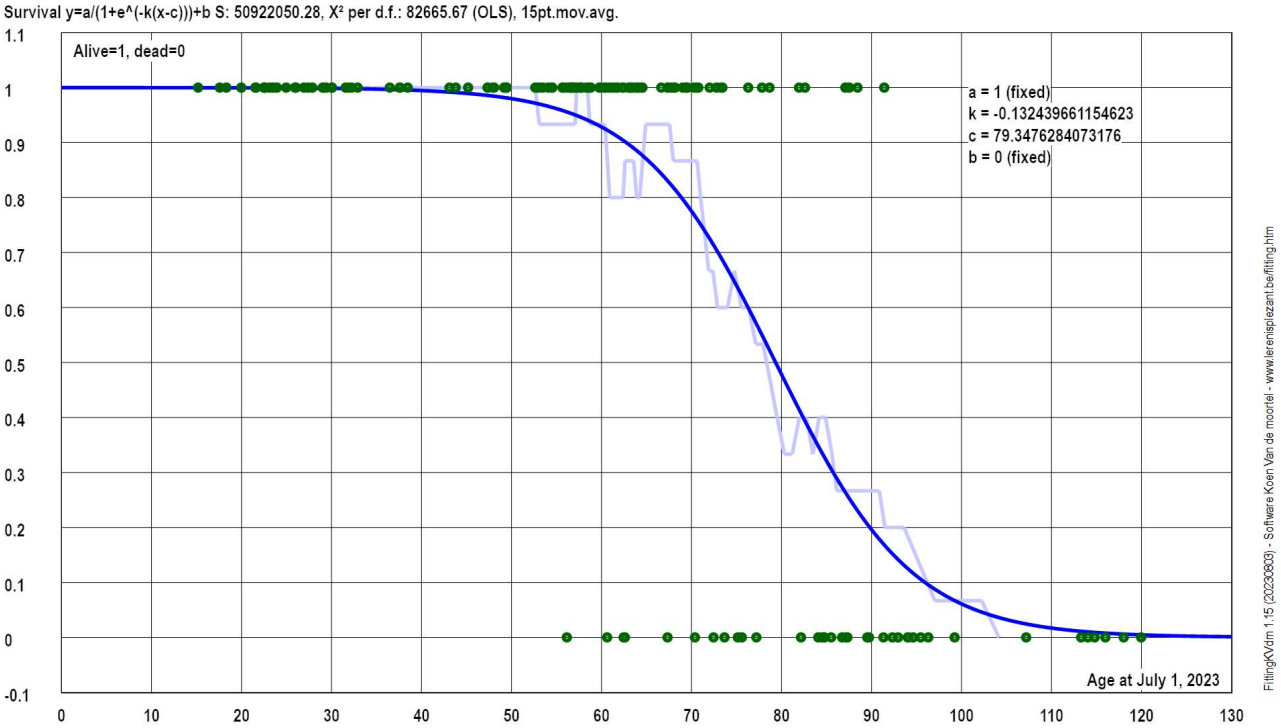
Also the “Weibull decay” model fits reasonably. Set a = 1000000 because that’s the initial number of people.



*** Survival_friends-family.dta1**

Variables: x = time since birth in years at July 1, 2023, for 156 people I know/knew personally; y = 1 if the person was alive, y = 0 if the person was dead.
Using the 0/1 y values instead, there is no "measurement error", so one has to enter some artificial fake value like 0.001 (small enough to make the error flags invisible).

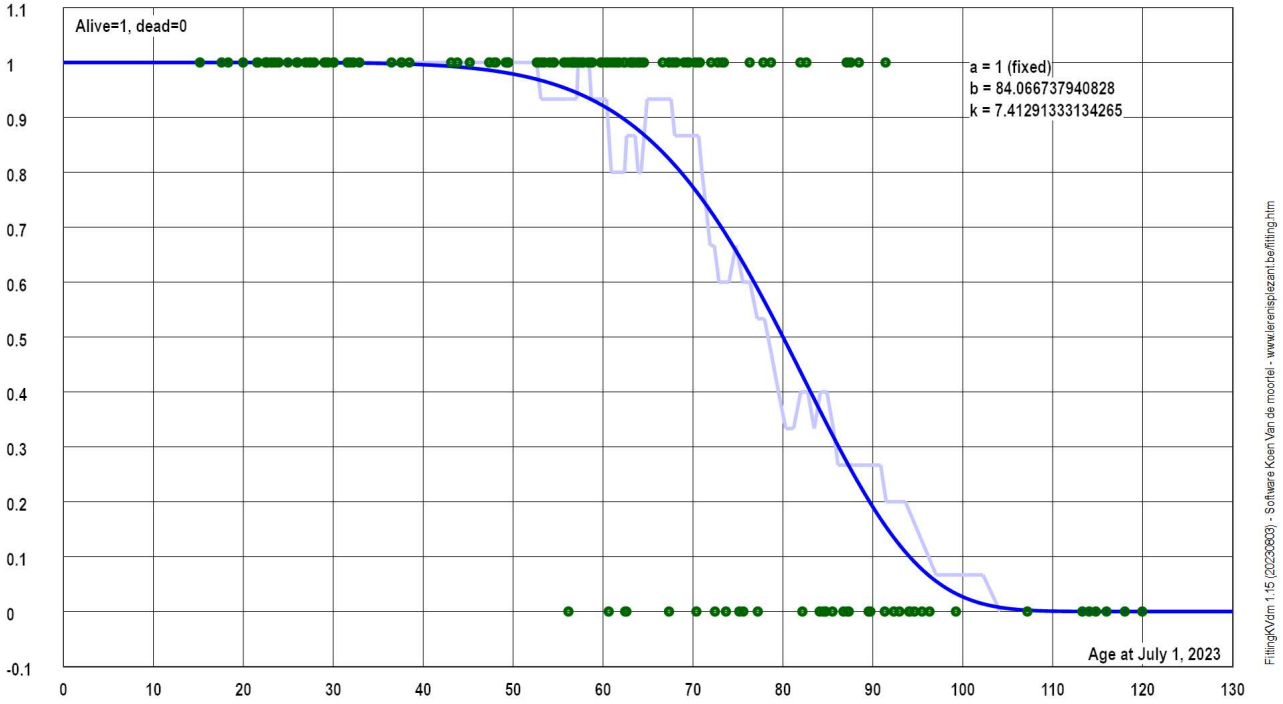
This is a typical example of the "mis-use" of logistic models.
What we want is "the probability of being alive after x years". That probability could be estimated for example by calculating the average (and standard deviation) for the y values in a number of intervals, but then you would need more data because you throw away information. It is usually *assumed* that these interval averages or a moving average can be approximated by the simplest sigmoid curve: the logistic function, with lower limit 0 and upper limit 1.



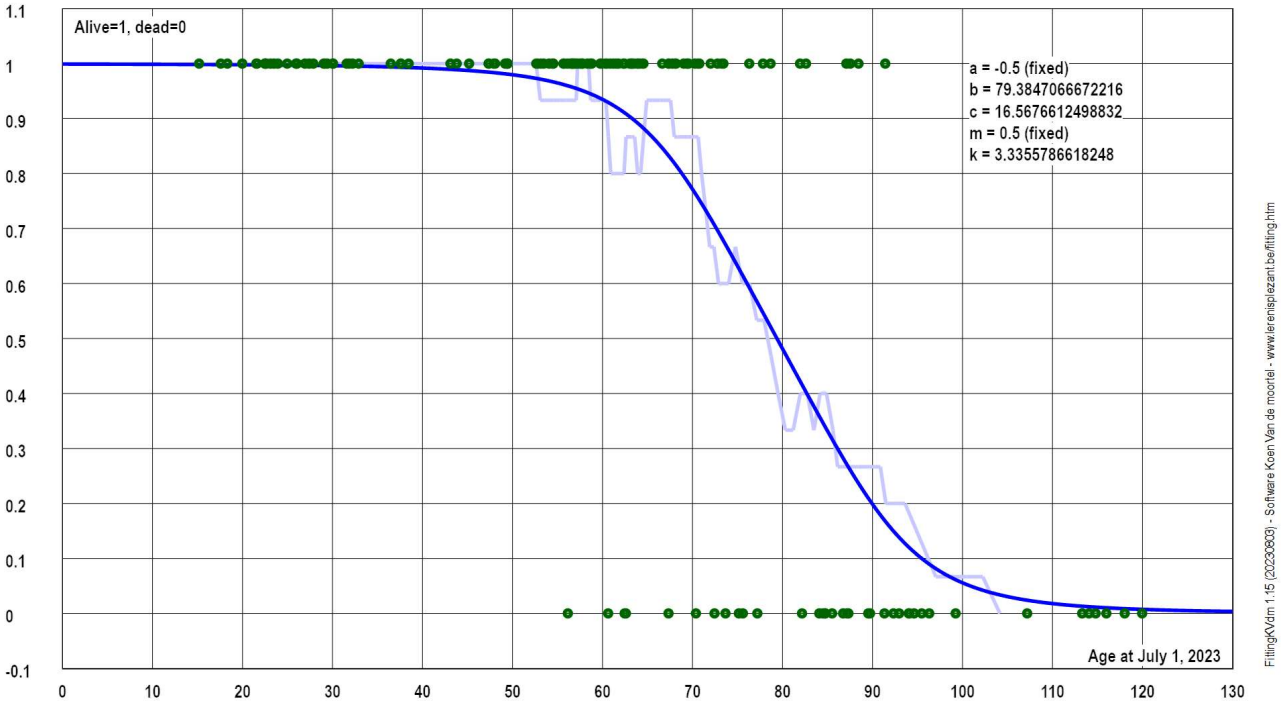
From this fitting we can estimate that half of the people die before the age of 79.3 years.
The curve follows more or less the 15 point moving average (click "More options" in the graph section to set this).

Other models, like "Weibull decay" (with upper limit a=1), and "Transition" (with a=-0.5 and m=0.5), also looks plausible, but it's impossible to distinguish which one is the "best fit", because we have only two possible y values!
The inflection point is also approximately (80, 0.5).

Survival y=a·e^(-((x/b)^k)) S: 50900165.5, X² per d.f.: 82630.14 (y), 12457.44 (x) (OLS), 15pt.mov.avg.



Survival y=a(x-b)/((x-b)^k+c^k)^(1/k)+m S: 50900794.29, X² per d.f.: 83171.23 (OLS), 15pt.mov.avg.

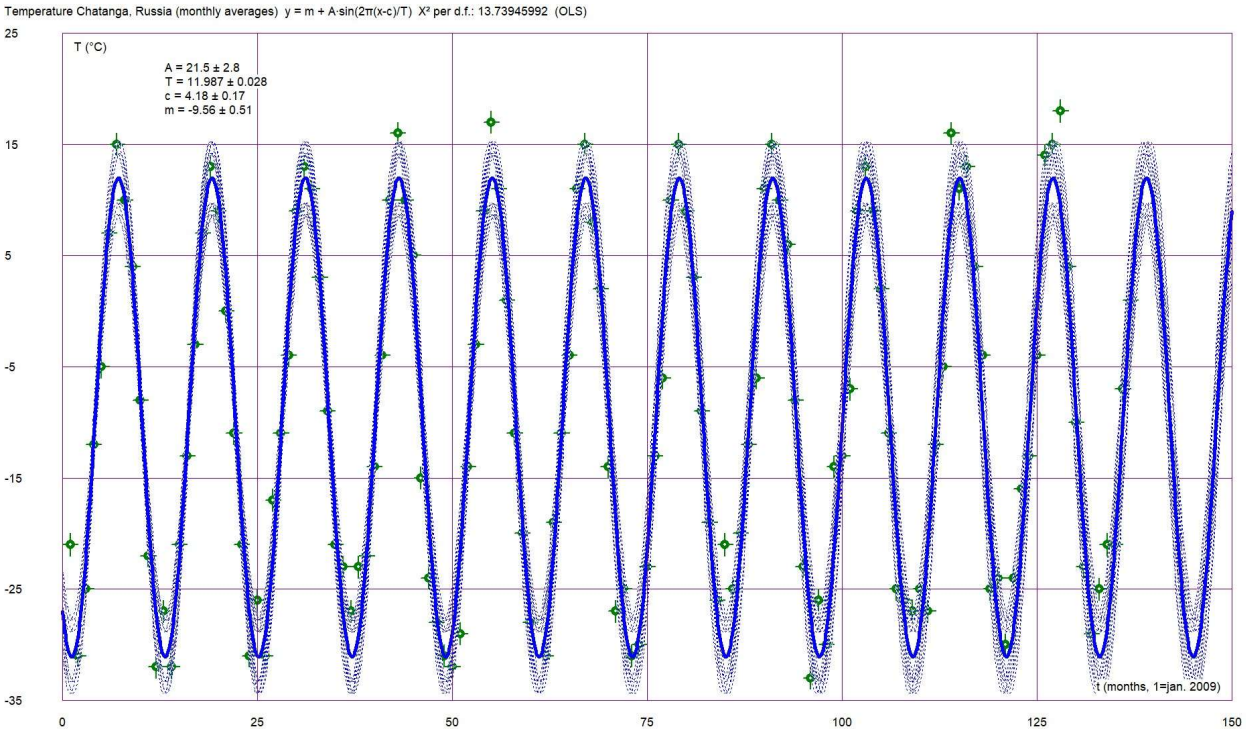


* Temperature Chatanga.dta1

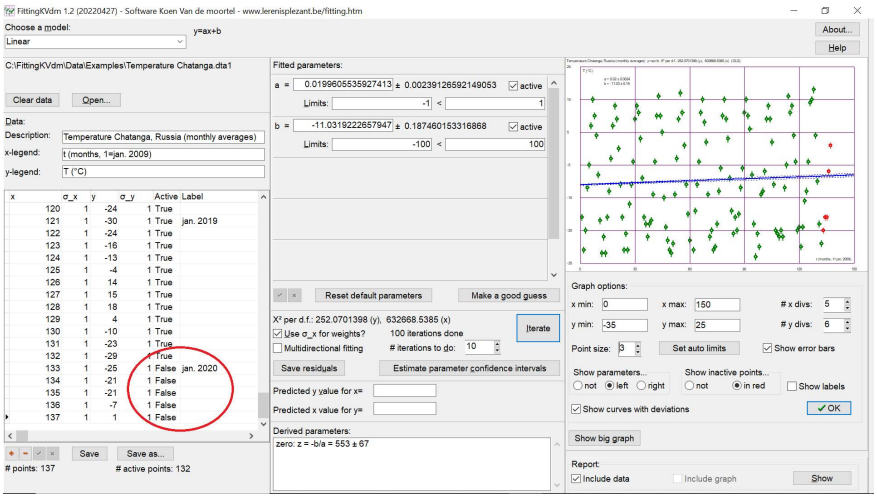
Variables: x = time (t, in months starting from January 2009, the last is May 2020); y = the average temperature of that month (in °C), in Chatanga, Russia.

Source: www.worldweatheronline.com/chatanga-weather-history/taymyr/ru.aspx

Of course, a "Sine wave" model will make a lot of sense here!
Just click on "Make a good guess" and some iterations will produce good results. Notice that you might expect a period of exactly 12 months, and the algorithm produces 11.987 which is not bad of course. For more precision, you have to use the exact number of days since a certain start date (see: sun spots), since some years have more days than others! Also try the "Skewed wave" to detect if there is a systematic asymmetry between the "heating up" and the "cooling down" times of the year.



Of course, you could also apply the "Linear" model here to find out if there is a trend towards warming or cooling. But, make sure the data cover an integer number of years (a multiple of 12 months) then, by setting the last data points to "inactive"!
In the given time span, you will find a warming of 0.02°C per year.

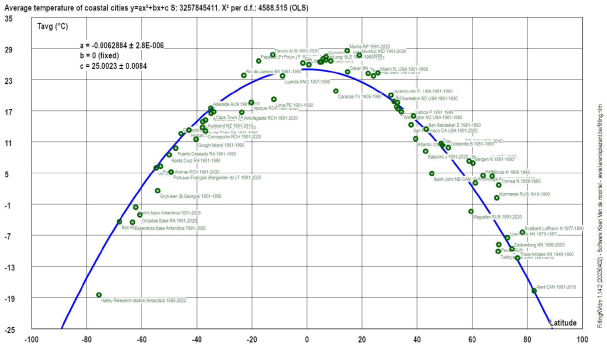


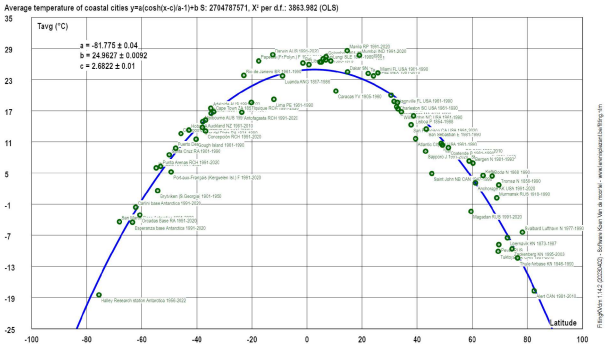
* Temperature in coastal cities.dta1

Variables: x = geographical latitudes of coastal cities; y = average daily temperature in °C.
The measurement imprecisions were assumed to be half of the least significant digit.
The time span for averaging the temperature was not the same for every location, which will unavoidably cause some "noise".

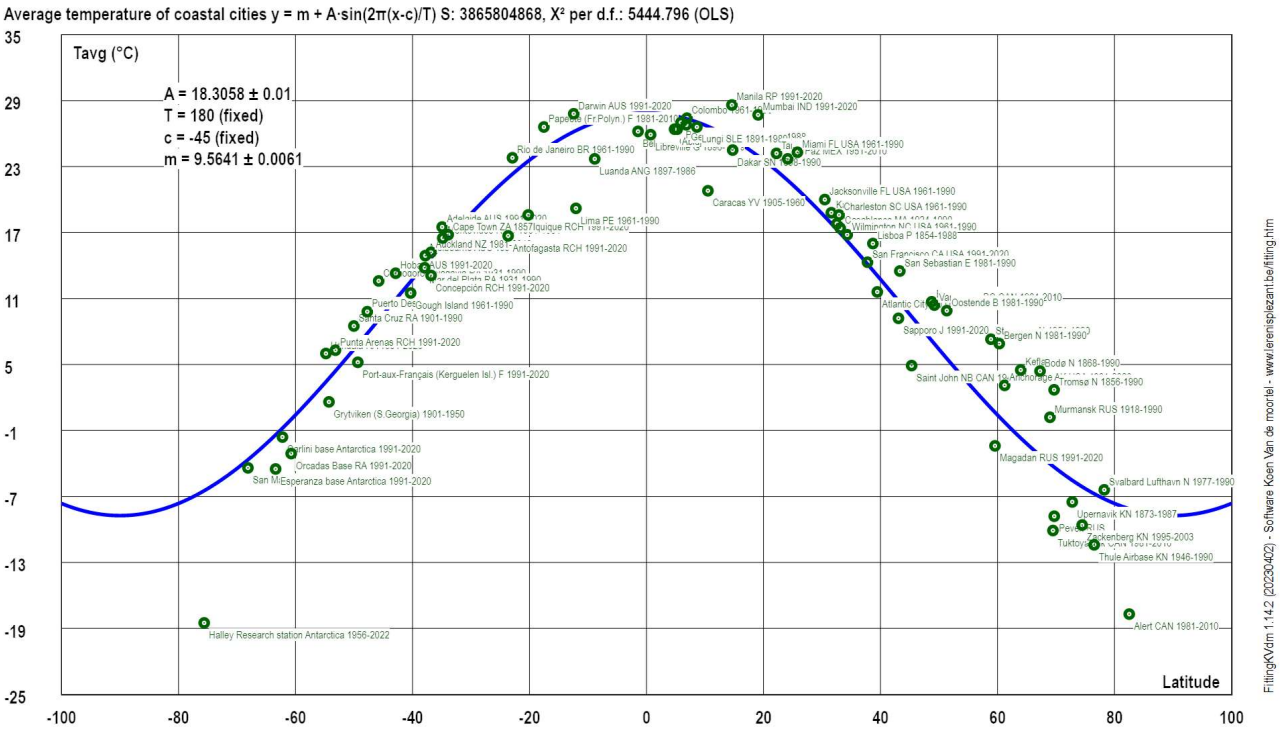
Sources: www.worldclimate.com and https://en.wikipedia.org

Which model can we expect to work here? You can find some weird "solutions" for this on the internet, like here: http://webinquiry.org/examples/temps/ where a mess of linear pieces is used!
One might think that a parabole (left) fits well, or a chain line (right), but the extrapolation makes no sense!





We better look at the underlying physical reality, even a simplified version. If we assume that the earth is a ball flying at a constant distance from the Sun, and the average temperature is proportional with the average insolation T_{avg} should be more or less proportional to the cosine of the latitude/2 (= sin(2π(x-45)/180). The truth is more complicated but this is fairly close; see "Global Physical Climatology" 2nd ed., by Dennis L. Hartmann, Elsevier 2016.



It might seem that the fit is worse for extreme latitudes, but actually there is a good reason for that: the Haley Research Station at the Antarctic coast and the base called "Alert" in northern Canada are most of the year surrounded by ice, so their winters are more like places with a land climate; they don't benefit from water tempering the temperatures like on other coastal places!

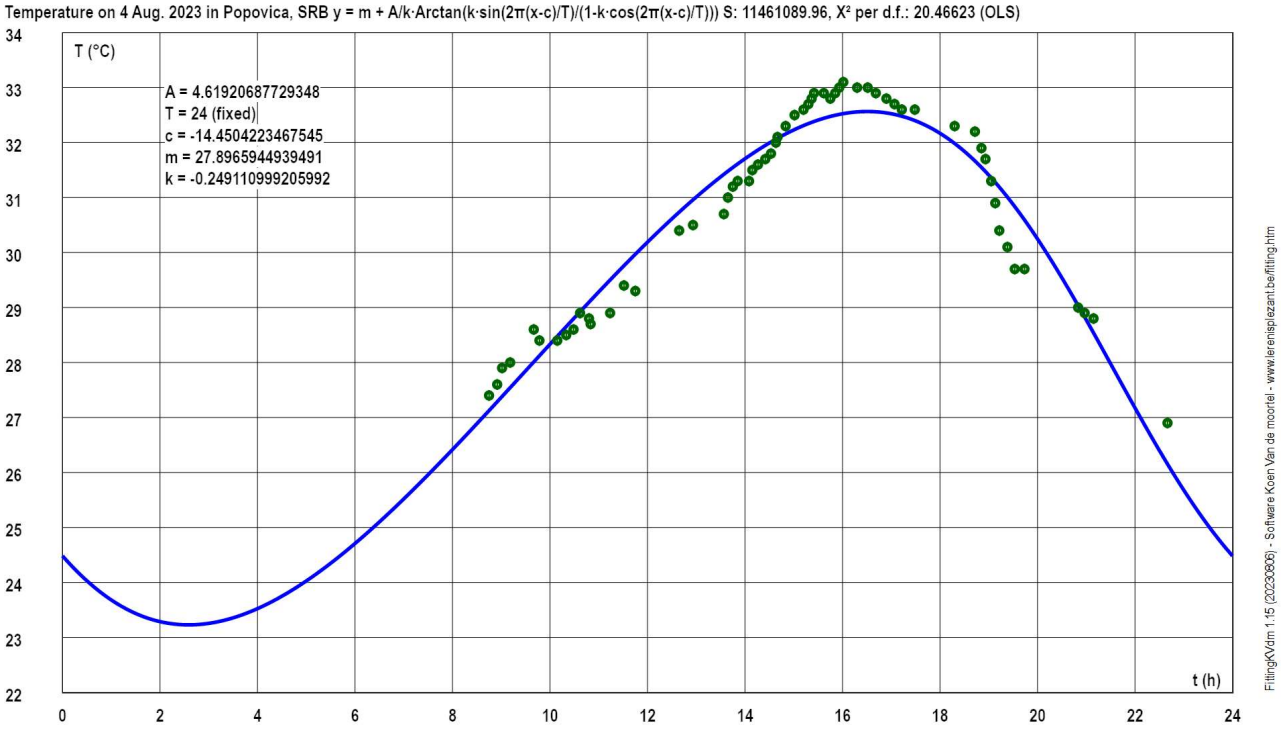
It's interesting to see that places near a cold sea current, like Lima (Peru), are way below the curve, while those near a warm current, like Bergen (Norway), are above it.

* Temp Popovica 20230804.dta1

Variables: x = clock time (decimal hours); y = temperature (°C) measured 1m above the ground, in the shade, in the village of Popovica, in the hills a few km south of Novi Sad, Serbia, on 4 August 2023.

Source: own measurements. The weather on that day was quite stable and sunny. The thermometer had a resolution of 0.1°. The absolute precistion was not known, but by comparing with other thermometers we could assume a precision of 0.1°C.

The simplest model to use here, would be the "Sine wave". Fix T to 24, since there are not enough data points to auto-detect the period precisely.
There seems to be an asymmetry though: the warming up goes slower than the cooling down, so you might try the "Skewed wave".

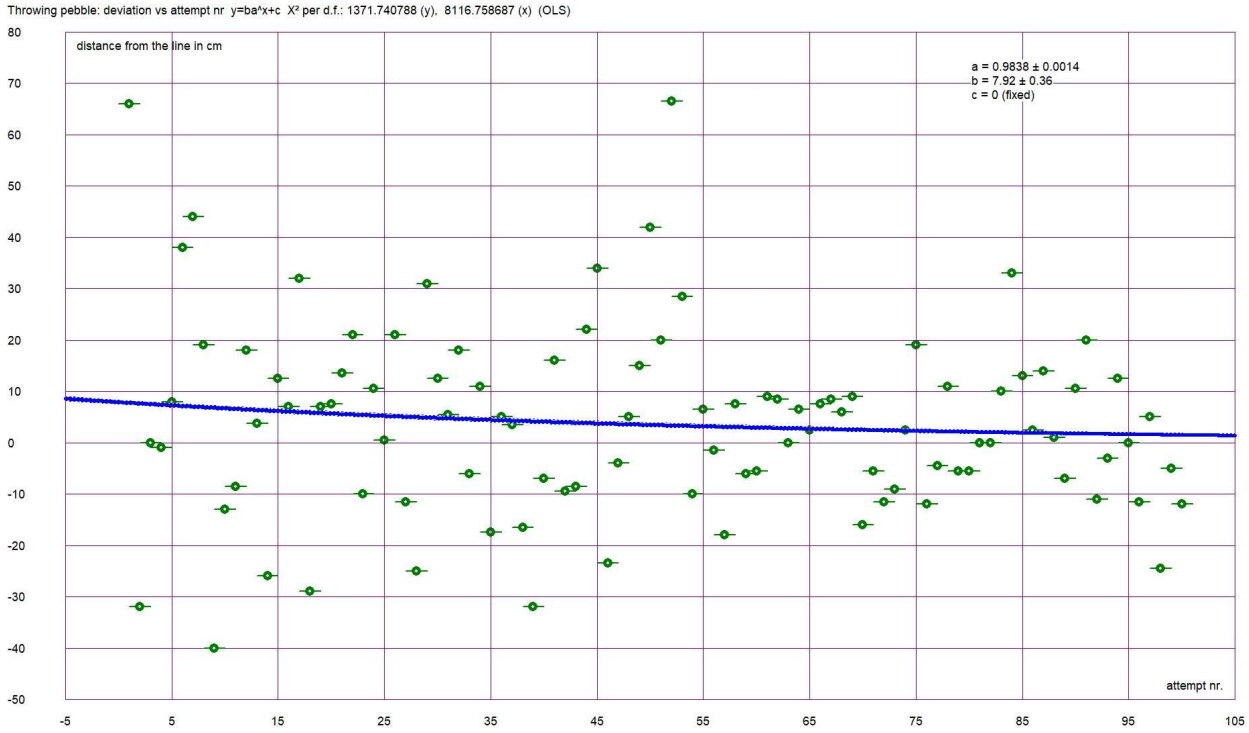


The "Sine wave with harmonics" will fit "nicer", but... it predicts a night temperature of 1°C, which is very absurd here. This is a good example of "overfitting": the model adapts to accidental fluctuations, and that's not what we want.

* Throwing a pebble.dta1

A pebble was thrown 100 times to a marker line and each time it was measured how much the test person was wrong.
Variables: x = attempt number, y = distance in cm between the center of the pebble and the line (+ = too far, - = not far enough). Since the attempts were done consecutively, x is closely related to the time of course.

The interesting thing here, was a **learning effect**. This can be measured by fitting with the "Exponential + baseline" model with a fixed c=0 (no base line since the ideal limit is 0).



More about this experiment:
https://www.researchgate.net/publication/360454548_Measuring_psychomotor_improvement_-_a_practical_example

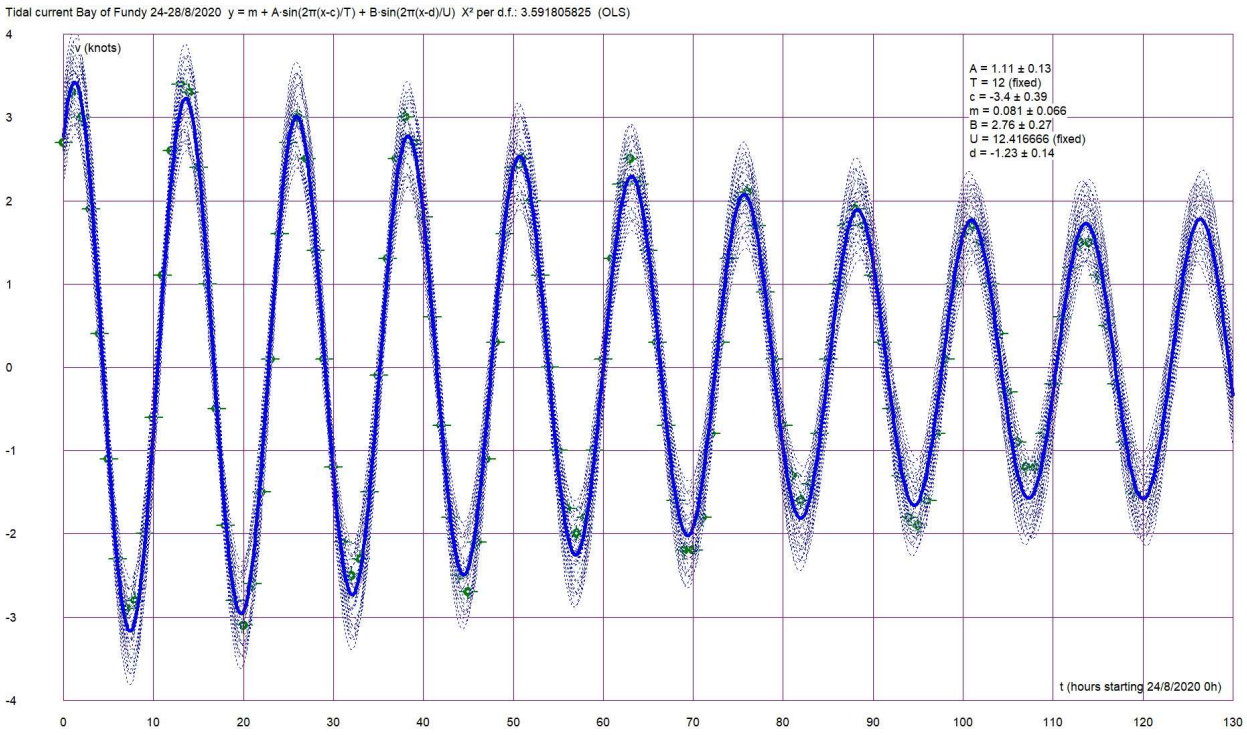
* Tidal_current_Fundy.dta1

Variables: x = time (t) in hours starting 24 August 2020; y = speed (v, in knots) of the tidal current at the entrance of the Bay of Fundy, Canada.

Source:
l-36.com/tide_week.php?location=Grand%20Manan%20Channel%20(Bay%20of%20Fundy%20Entrance),%20New%20Brunswick%20Current

Try the "Added Sine wave" model to find the pattern!

Now, here we can make the convergence much easier because we know the two periods: T = 12 (from the Sun) and U ≈ 12h25min ≈ 12.41666…h (from the Moon), so enter them to start and set them "inactive". The fit is good, but not perfect, since the exact movement of the Moon is very complex!



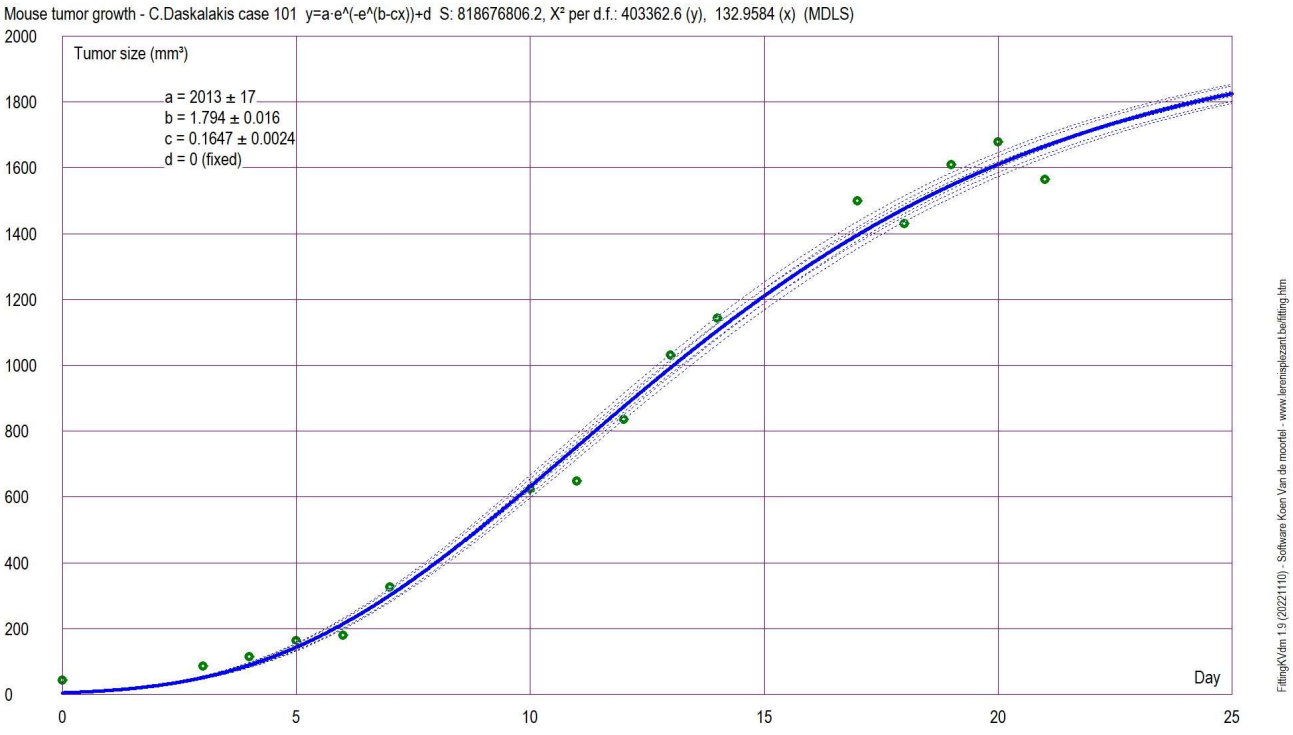
* Tumor growth Daskalakis 101.dta1

Variables: x = time in days; y = size of a tumor in a mouse (in mm³).

Source: Constantine Daskalakis, "Tumor Growth Dataset", TSHS Resources Portal (2016), www.causeweb.org/tshs/tumor-growth, case 101 (no treatment).

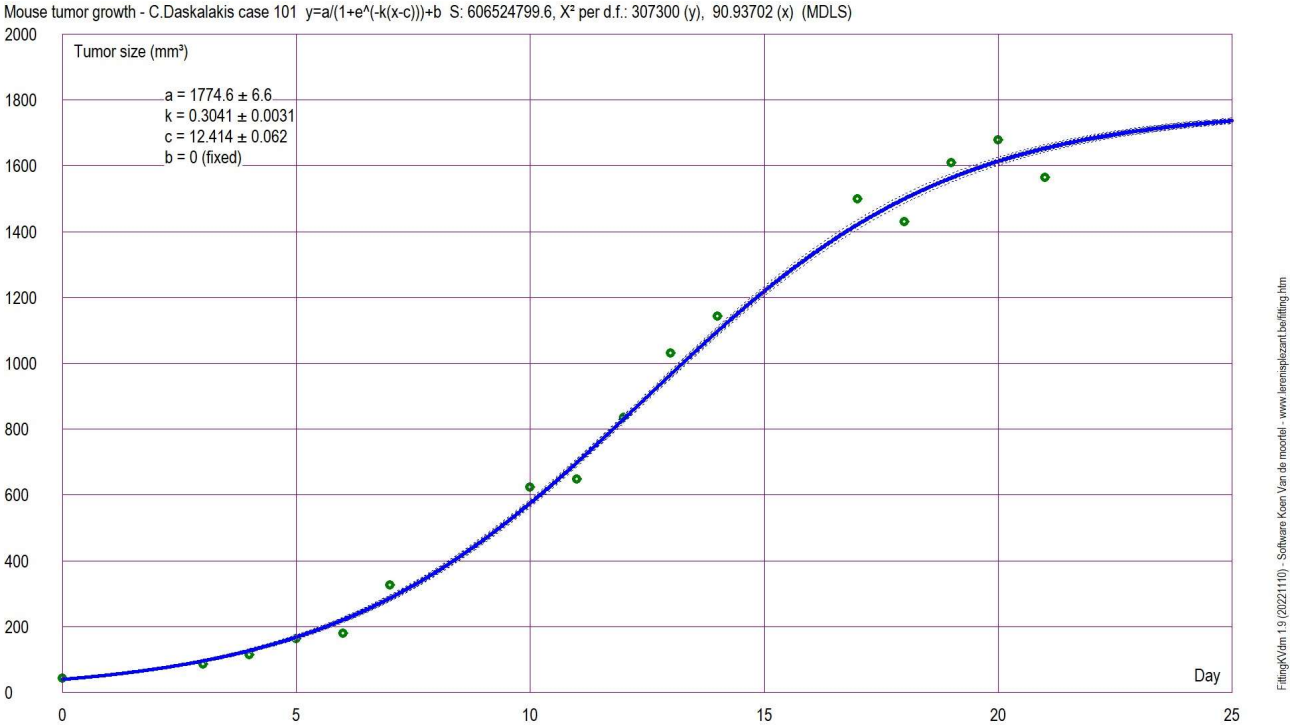
According to people in this field, this kind of growth can be described quite accurately with the Gompertz model function (without baseline, so d = 0). Parameter a predicts the maximal size the tumor will get, and c the speed of the growth.

Since the limit value (the finally expected tumor size) is far above the last measurement, you can use MDLS. You should enter a starting value for a, higher than the measurements (e.g. 1800) then. OLS and MDLS don't differ very much in this case.



Notice that also the "Logistic" model fits well (with b = 0), actually even better (smaller S). But it predicts a smaller final size (1774.6 in stead of 2013).

Repeated comparisons of predicted and observed values might sort out the best empirical model for this phenomenon.

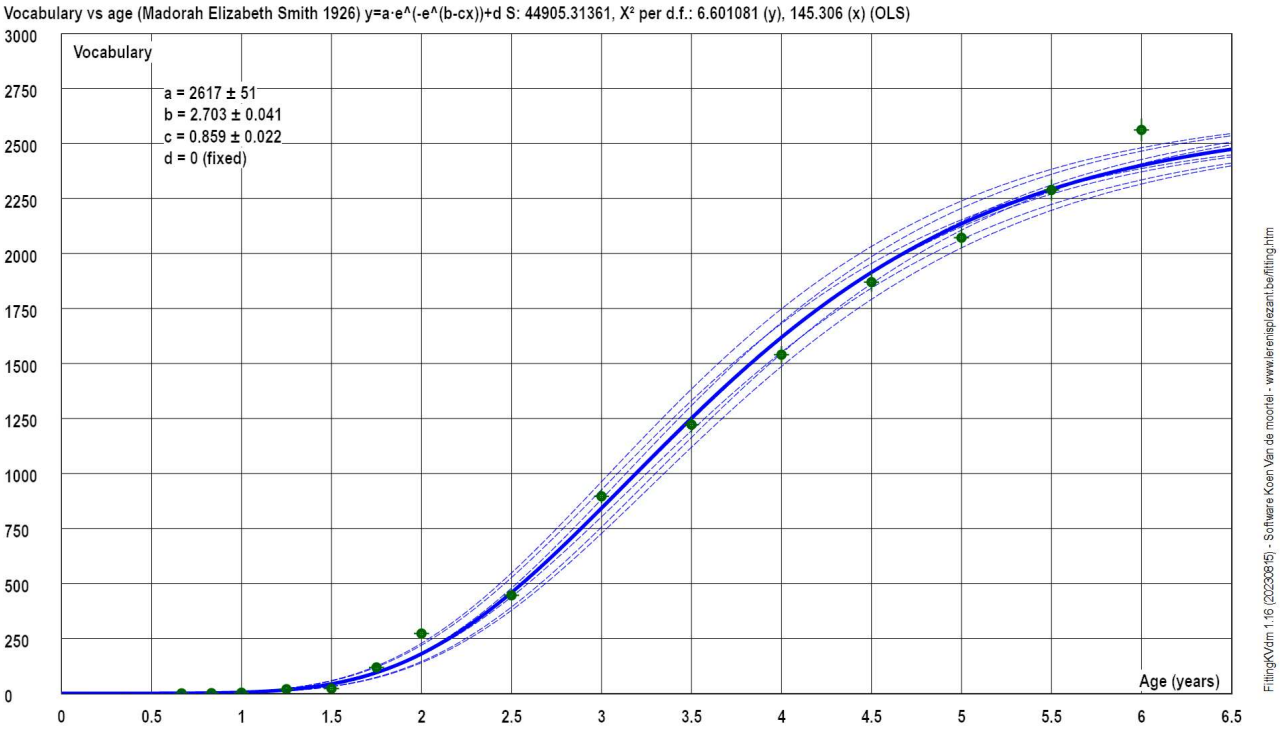

Mouse tumor growth - C.Daskalakis case 101 y=a/(1+e^(-k(x-c)))+b S: 606524799.6, X² per d.f.: 307300 (y), 90.93702 (x) (MDLS)
Tumor size (mm³)
a = 1774.6 ± 6.6
k = 0.3041 ± 0.0031
c = 12.414 ± 0.062
b = 0 (fixed)
Day
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
5
10
15
20
25
FittingKVdm 1.9 (20221110) - Software Koen Van de moortel - www.lerenisplezant.be/fitting.htm

* Vocabulary vs age (Smith 1926).dta1

Variables: x = age in years (0..6), y = average number of words observed in the vocabulary of children of that age. The error on x is assumed to be half a month, on y: the square root.

Source: Madorah Elizabeth Smith: "An investigation of the development of the sentence and the extent of vocabulary in young children" (Iowa Child Welfare Research Station, vol.III nr.5, May 1926), p.52. Can be found online:
pure.mpg.de/rest/items/item_2385505_3/component/file_2464108/content

This seems to be a tricky data set to model. We would expect a growth curve that predicts approximately the vocabulary of an average adult as the limit, but typical models for this, like "Weibull growth", "Logistic" or "Transition", predict impossibly low values.
Only "Gompertz growth" (with d=0) produces something that looks reasonable, but it also predicts a limit of only 2616.7 words.
So this model might be used as an estimation for age 0 to 6, but not much further!

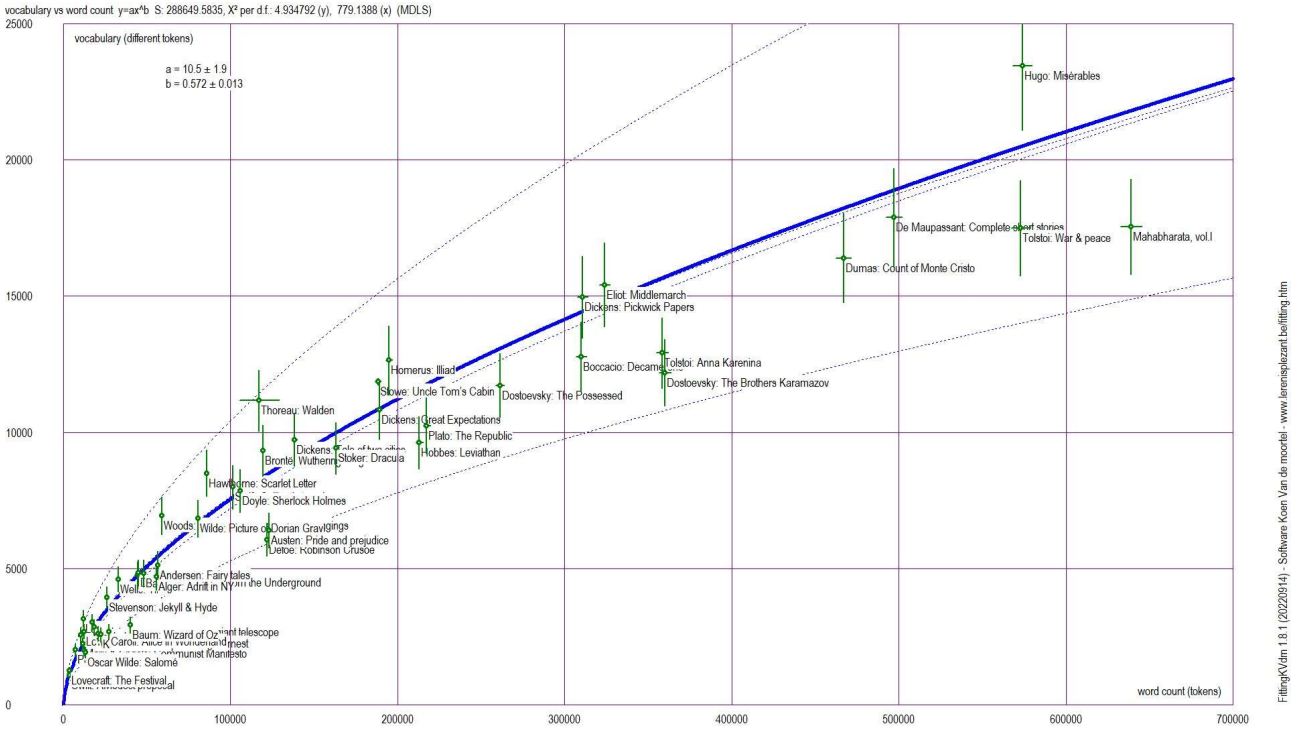


* Vocabulary vs word count.dta1

Variables: x = number of words (tokens) in a book; y = number of different tokens in the book.

Source: 51 books (in plain text format) were downloaded from:
www.gutenberg.org/ebooks
and analyzed with "QUITA" (Quantitative Text Analyzer 1.1.9.0); see: korpus.cz/quitaup.
Note: the numbers obtained are not perfect: some tokens were mistakenly seen for words, like "I" can be "me" or a roman 1, so a very rough error estimation of 1% was given. The error on the y values must be bigger, especially if you consider different versions of the same word (singular/plural, verb forms etc.) as the same. A rough estimation of 10% was given. Actually the y values should then all be somewhat less, but to remove all the duplicate word versions would be a very tedious task, something for fanatic linguists, not essential for our example here. If we just want to compare writers, it doesn't matter if all the values have the same systematic error.

Which model is appropriate here? One thing is sure: the curve has to go through the origin: 0 words = 0 different words! The curve should be ascending, but to infinity or is there an upper limit? There is no book with an infinite number of words and new words can always be invented, so maybe there is no real limit but the sky.
It seems that empirically, a power function fits quite well. There is even a name for this phenomenon: **"Heaps' law"**, see: en.wikipedia.org/wiki/Heaps%27_law.

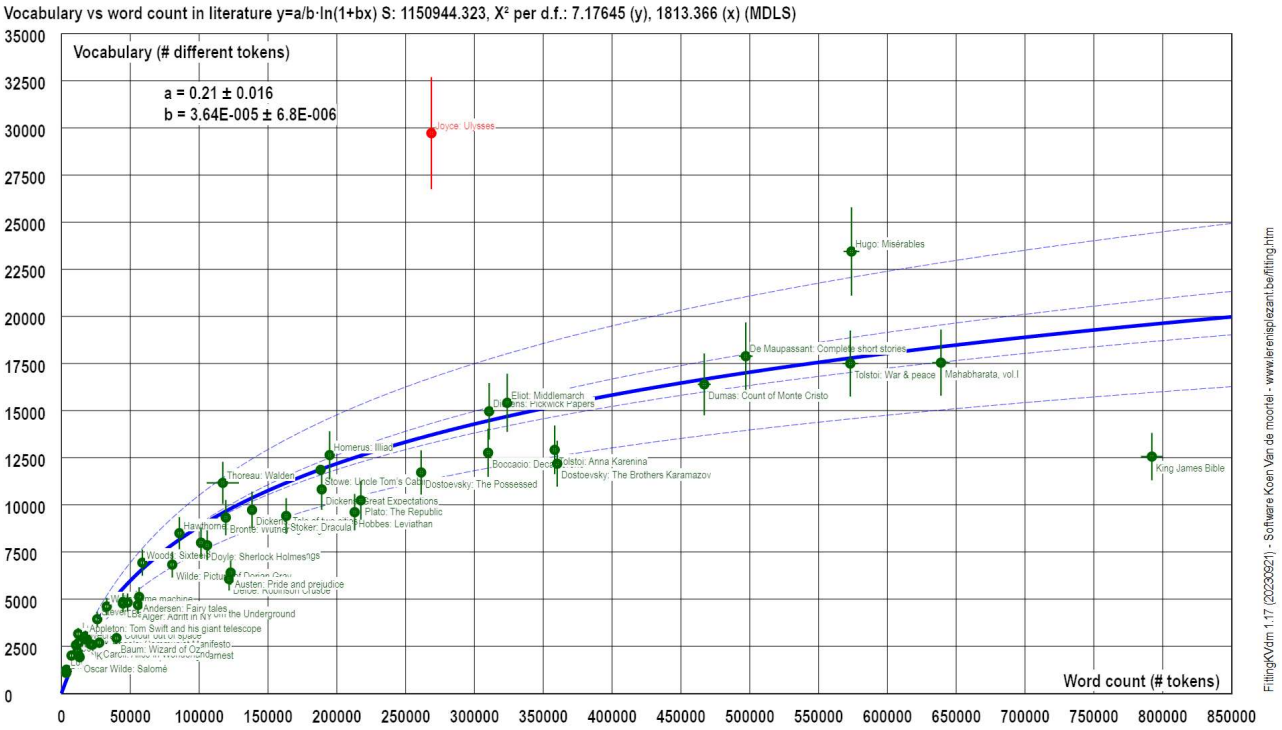


Normally I am not in favour of leaving out so-called "outliers", but "Ulysses" by James Joyce (reputed as an incomprehensable book) is so far above the rest, and the King James Bible translation so far below, and they both have a strong leverage on the curve, so it might be wise to leave them out if you want to see the general pattern. Or, you could add several hundreds of other books to the data file, and then the leverage of these

two “abnormal” books would automatically be less.

You can use this model to judge the vocabulary of the authors: the more above the curve, the larger it is. If you want one “vocabulary rating”, you might do this fitting for each author separately and, for example, calculate the vocabulary if the author would write a 10000 word book. The value expected by the power model is 2024.32.

There is one principal problem with the power function though: it’s infinitely steep in the beginning, which means that for every word, an infinite number of *new* words would be added. Although the graph looks globally nice, that’s an ugly detail!
Another model function does NOT have this problem: the “Logarithm shifted through 0”! It fits nicely, even with the bible included, and f’(0) = 0.21! That makes a lot more sense!



---

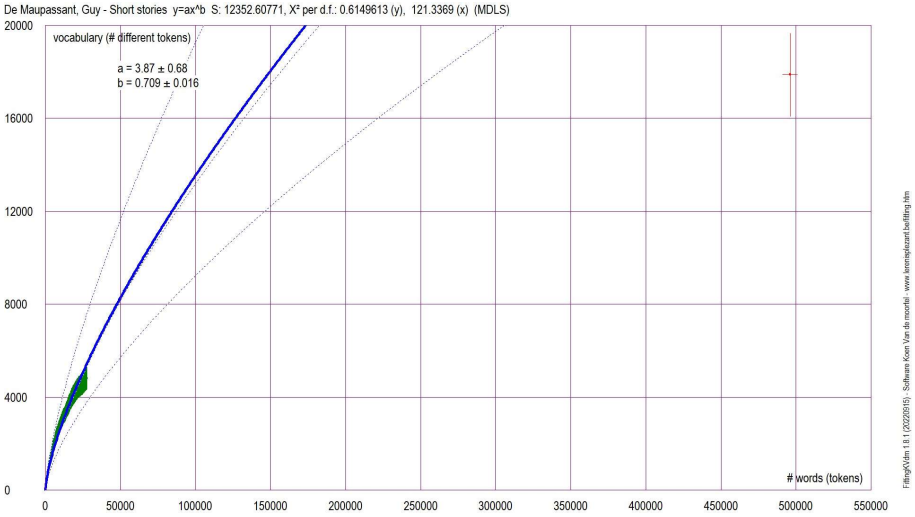
* Vocabulary vs word count - De Maupassant-Short stories p1-80 & all.dta1

Variables: like the previous example, but the numbers come from 1 book: “Complete Original Short Stories” by Guy De Maupassant (translated in English). For i = 1..80: x_i = number of tokens in the first i (screen) pages, and y_i = number of unique tokens in those pages. The last x and y are for the whole book.

Source: www.gutenberg.org/files/3090/3090-0.txt

In this case, we get strictly ascending rows of x and y values: with every page x and y increase, but y lesser and lesser, because the percentage of new words on every new page descends. The curve is now much less chaotic of course.

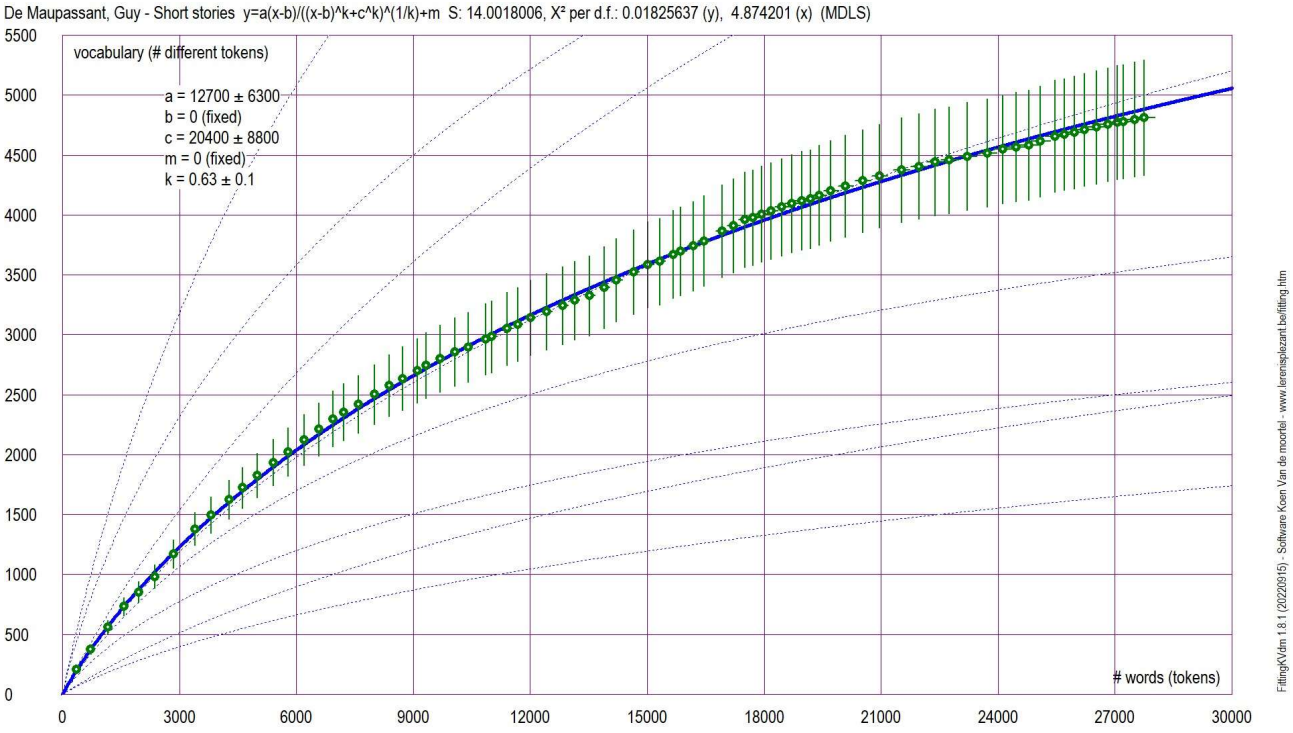
The power function seems to fit reasonably, but… the last point is a good test to see how well this model predicts! If you fit without including that last point, the model will seriously overestimate the vocabulary of the total book (point above right)! Using OLS it's even much worse (OLS: 70980, MDSL: 42121, reality: 17882).



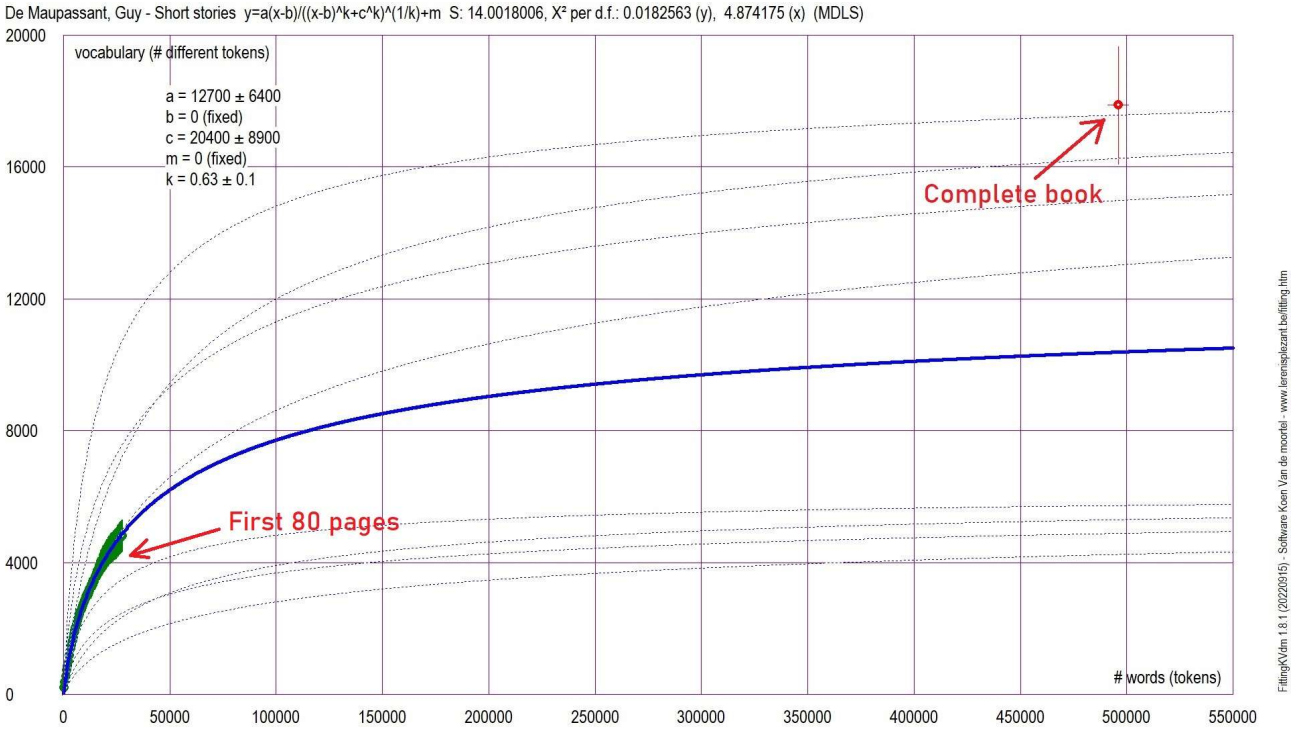
The vocabulary for a 10000 word book predicted by the "Power" model would be 2647, which makes de Maupassant a "rich" writer (see previous example file).

If you assume there might be an upper limit, you can try "Weibull growth", "Rational 1", or "Plateau-exponential" with c = 0, but they seriously underestimate the total vocabulary. Only the "Transition" model with b = m = 0 actually fits nicely, better than the "Power" model, although the asymptote stays too low too (12700 instead of 17882). This is still closer to reality than the prediction from the "Power" model (42141).

The "Transition" model fitting looks amazing:



With expanded scale:



Using OLS, the limit value is (accidentally?) a bit higher, but the χ^2 values are 10 times as much! (Don't compare the S values, because they are calculated differently.)

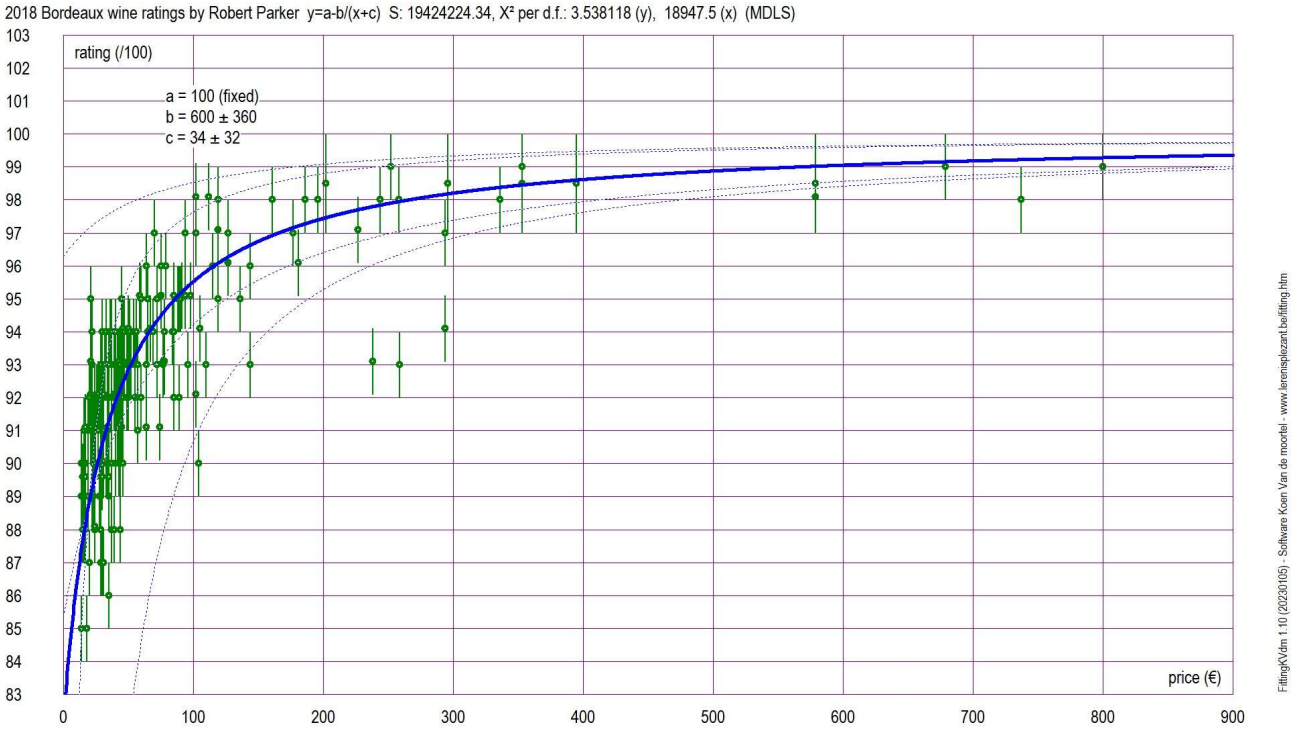
The "Logarithm shifted through 0" model looks very similar, but it comes closer to the end value, and better with MDLS than with OLS.

*** Wine ratings vs price Bordeaux 2018 RP.dta1**

Variables: x = price of a 2018 Bordeaux wine (in €) when it came out, y = rating by Lisa Perrotti-Brown for robertparker.com (on a scale of 0..100). A rating of "95-97" was entered as "96±1"; "95-97+" was converted to "96.1±1".

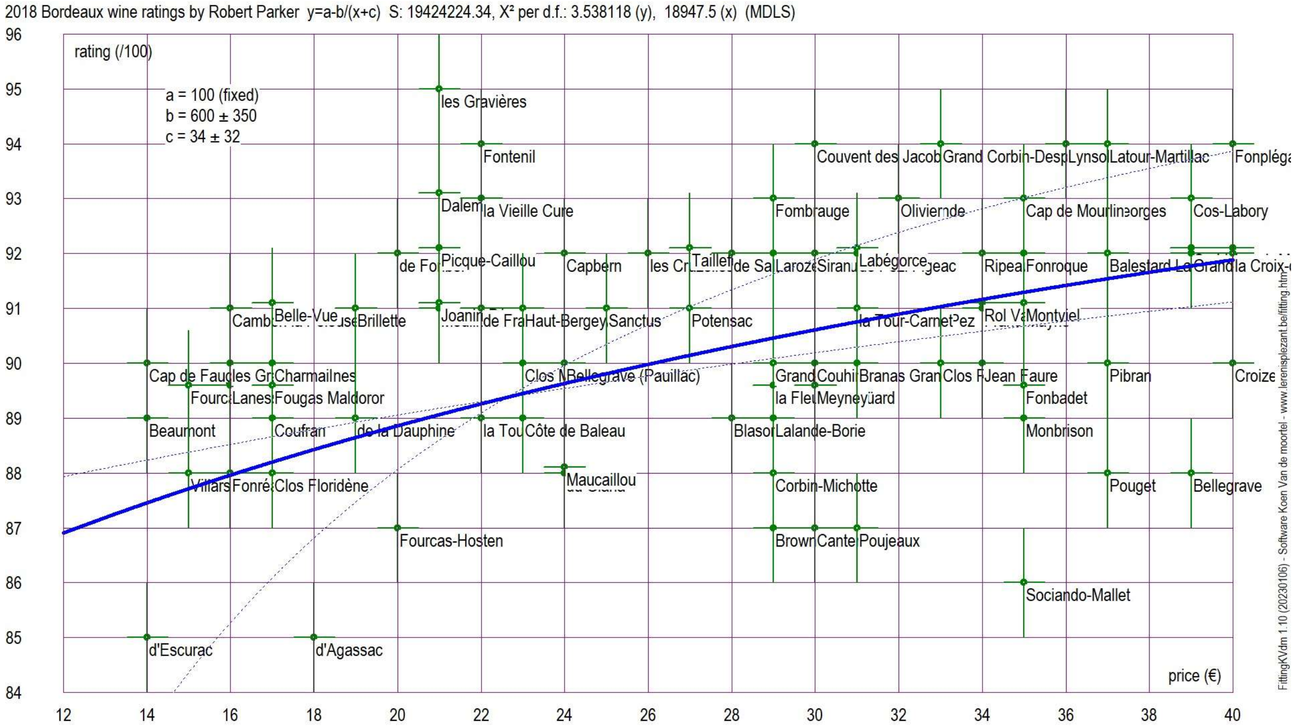
Source: https://www.bordoverview.com/?q=Robert-Parker

The (subjective) quality of a wine is certainly not nicely proportional to the price. A model that reflects more or less the relationship needs to be ascending with a horizontal asymptote since there is a limit value (in this case: 100) if the prices go to infinity. It doesn't have to go through the origin. The "Rational 2" model fits reasonably.



Strangely, a "free" wine would still get 83 points… The "Rational_1" model, with a = 100, that goes through the origin, might make more sense perhaps?

What's the use of such a model? Well, suppose you want to choose a wine with a good quality for the price, you can scale the graph according to your budget, switch the labels on, and look which wines are far above the curve. E.g.:



"Les Gravières" and "Fontenil" seem to be excellent, and "d'Agassac" and "Sociando-Mallet" seem to be disappointing for the price, according to Lisa Perrotti-Brown. Strangely enough, the latter are a few of my favorite wines, so apparently Lisa doesn't have the same taste as me... You could do this exercise with ratings from other tasters and find out whose advice you like to follow the most.

* Wine rating comparison.dta1

Variables: x = 2018 Bordeaux wine rating by Lisa Perrotti-Brown, y = rating by Tim Atkin.

Source: https://www.bordoverview.com/?year=2018&bank=both

Tastes differ, even among professional tasters! On a scatterplot of the judgments of both sommeliers we don't really see a line, but rather a cloud. There is some correlation, but Kendall's τ is only 0.356 (click the "Show" button in the "Report" panel to see this).
The best fitting line can be obtained with the "Linear" model, obviously. Certainly use MDLS here, because there is absolutely no preference for choosing Tim as x and Lisa as y or vice versa; in both cases we should get the same line (mirrored). Using OLS this is NOT the case.