Fitting mortality functions

2022 09 10 Bob Howard

Revised: 2022 09 15

# Background

It is often thought that there must be some underlying mathematical relationship that governs mortality. If we knew what that relationship was, it would be relatively simple to develop a mortality table by fitting a function[[1]](#footnote-1) to observed experience.

There are three main reasons for finding a mortality function. In my work I have used it mostly for the second reason.

1. One is convinced that mortality follows a specific function.
2. There is a need to extrapolate mortality rates over ages for which there is insufficient quantity or reliability of data. The function is fit over a range of ages with adequate data, and then extended to the ages needed.
3. There is not a sufficient quantity of data to use other methods, but there is a need to determine mortality rates at each age over a fairly wide range of ages.

In practice fitting a function is not simple. No one has yet found a function that fits well over a wide range of ages. However, there have been some that have proved useful at higher ages.

Benjamin Gompertz proposed a relationship in 1825 which is still regarded as appropriate by many.

or (1)

The form on the first is the one more commonly displayed, but the second is more useful for our purpose.

The Kannisto function was suggested by Väinö Kannisto in 1994 in order to take into account the observed mortality deceleration at advanced ages, and it is often used in recent mortality projects.

(2)

Gompertz’ function is an exponential. Kannisto’s function is a standard logistic. Because the asymptote of a standard logistic function is 1, one might also see it used as a probability function, as in

(3)

# µ or q?

Mortality functions usually specify the force of mortality. In practice, an actuary is likely not interested in the force of mortality but in developing an appropriate set of mortality rates from observed experience or in extrapolating mortality rates to ages beyond those available in reliable experience.

The starting point for the work is likely an experience study. It is very difficult to structure an experience study to derive the force of mortality directly. In most cases the result is either a mortality rate or a central death rate. Actuaries tend to prefer the former, and demographers the latter. Being an actuary, I will skip any further reference to central death rates.

A common approximation to force of mortality is

(4)

This is easily applied because the raw mortality rates are readily available. The approximation assumes that *µx* is linear between *x* and *x*+1. The approximation is accurate only to about three significant digits at most ages, and that it can be quite inaccurate at very high ages where the force of mortality may be changing rapidly. However, it is probably still acceptable because the uncertainty in the mortality rate is much larger than the third significant digit and because raw mortality rates over age 100 are not likely to be used because they are almost certainly unreliable in any insurance or annuity study.

But what if we have a function of the force of mortality and want the mortality rate?

The inverse of (4) is

(5)

But that expression is no better than (4) for accuracy, and if we have a function obtained by a fit to the data for a number of ages, we need something that is more accurate.

The generalized function is

(6)

Applying Simpson’s Rule will yield at least five significant digits of accuracy, and it gets better at very high ages. This will almost always be sufficient, and striving for more would fail to recognize the uncertainty inherent in the underlying experience data. Thus,

(7)

By the way, exact equations exist for both Gompertz and Kannisto to calculate *qx* from *µx* but not necessarily for other forms.

# Generalized expressions

Some actuaries have modified the Gompertz and Kannisto functions to provide for more parameters in hopes of a better fit over a longer range of ages. I have generalized these below while staying with polynomial exponents.

The generalized exponential form is

(8)

where *m* and *n* are integers and *m* < *n*, and the term *a0* is present unless both *m* and *n* have the same sign. The terms of the exponent represent successive powers of *x*.

The generalized logistic form is similar:

(9)

That is not to say that more terms in the exponent will give a better result, but one is obligated to test a variety of cases.

By the way, Makeham’s function is not included here because its parameters must be found by a different method than one used here.

# Solving for parameters

Regardless of the number of parameters, they can be found by minimizing a weighted least squares expression. The solution is illustrated below for the exponential case with *m* = -1 and *n* = 2. (Those values are used solely for illustration, not suggesting that it is appropriate to use those exponents in any particular case.)

Assume that raw mortality rates are known for each age from *xlo* to *xhi*; the summations are always over those same ages. Assume that (4) is used to approximate the force of mortality. Assume that the weights at each age, *wx*, are known and non-negative. I have generally used exposure for the weights, but thanks to Stefan Ramonat who pointed out that the fit is usually improved if the weights are death claims when the equations involve the force of mortality. (My testing agrees with him, but I didn’t find the difference in results from the two sets of weights to be very large. I have appended Stefan’s email at the end of this document for those who wish to consider this further.)

The expression to minimize, after taking logarithms of both sides, is

(10)

And the last term is approximated[[2]](#footnote-2) by

(11)

After taking partial derivatives with respect to each of the four parameters and setting the derivatives to zero, we get a system of four equations in four unknowns.

The solution for a logistic function is identical except that log is replaced by logit.

(12)

Thus, for the logistic function which is analogous to the above exponential function, we minimize

(13)

# Software

All of the above is fairly simple mathematics, and most actuaries could implement the method from the equations shown. But why should each actuary do that work with the attendant risk of error and need for careful checking? I have written a function, available in a dll, which accomplishes all the steps between raw mortality rates and rates that fit the function of the desired form.

MortFn(AgeLo, AgeHi, CalcLo, CalcHi, ExpLo, ExpHi, UseMu, UseLogit,   
raw, wt, q, stat)

where

AgeLo, AgeHi is the range of ages over which the parameters are fit

CalcLo, CalcHi is the range of ages for which mortality rates are calculated

ExpLo, ExpHi is the range of polynomial exponents used in the function

UseMu = 0 if the function is for mortality rates, otherwise the function is for the force of mortality

UseLogit = 0 for an exponential function, otherwise the function is logistic

raw is a vector of raw mortality rates for ages AgeLo to AgeHi

wt is a vector of weights for the same ages, probably exposures. The weights are first normalized so that they sum to AgeHi – AgeLo + 1.

q is a result vector containing the mortality rates for ages CalcLo to CalcHi. Note that the calculation always yields mortality rates, never forces of mortality.

stat is a result vector of length ExpHi – ExpLo + 3. The first item is the minimized value of the expression (involving log or logit). The second item is the sum of the weighted squared error in mortality rates. The remaining items are the parameters found when solving the system of equations.

Note that the first 8 arguments are all 4-byte integers, and the last 4 arguments are vectors 8-byte floating point numbers.

If the calculation is successful, the return value is 0. A non-zero value indicates an error.

The dll is available at <http://www.howardfamily.ca/graduation>.[[3]](#footnote-3)

On this same webpage there is an Excel workbook which demonstrates the use of the dll. Note that

1. You need to download the dll and place it somewhere in your path or specify the full path in the declare statement.
2. Include a declare statement at the start of your VBA module. It should be identical to my example except that you may use a fully qualified path following the “Lib” keyword.
3. When calling the function, the vectors which are the last four arguments must already exist and be the right length. The vectors are reference in VBA by indexing the first element of each vector. See my example.

To see what else you can do with WHGrad64.dll go to <http://www.howardfamily.ca/graduation>.

# References

Ahmadi, Seyed Saeed and Richard Brown. *Key Factors for Explaining Differences in Canadian Pensioner Baseline Mortality*. Canadian Institute of Actuaries; 2018. Available at <https://www.cia-ica.ca/docs/default-source/2018/218068.pdf>

Jordan, C.W. *Life Contingencies*. Chicago: Society of Actuaries; 1967.

Kannisto V. *Development of oldest-old mortality, 1950–1990: evidence from 28 developed countries*. Odense: Odense University Press; 1994.

# Email from Stefan Ramonat

Hi Bob,

I have been reading through your notes and other illustrations regarding fitting functions/curves/etc. to mortality experience, and there is a possible enhancement that you may want to consider. It is in relation to the weights used when applying linear least squares to log-, logit-, or cloglog- (the “complementary log-log”, log(−log(1 − x))) transformed q\_x’s. I believe you have been suggesting to use the exposures as the weight in such situations, but deaths as the weight will lead to better fits and should almost certainly be the preferred weight for the transformed data.

To illustrate, I fit a Gompertz function to some actual pensioner experience, where mu\_(x + ½) is taken as −log(1 − q\_x), under four approaches:

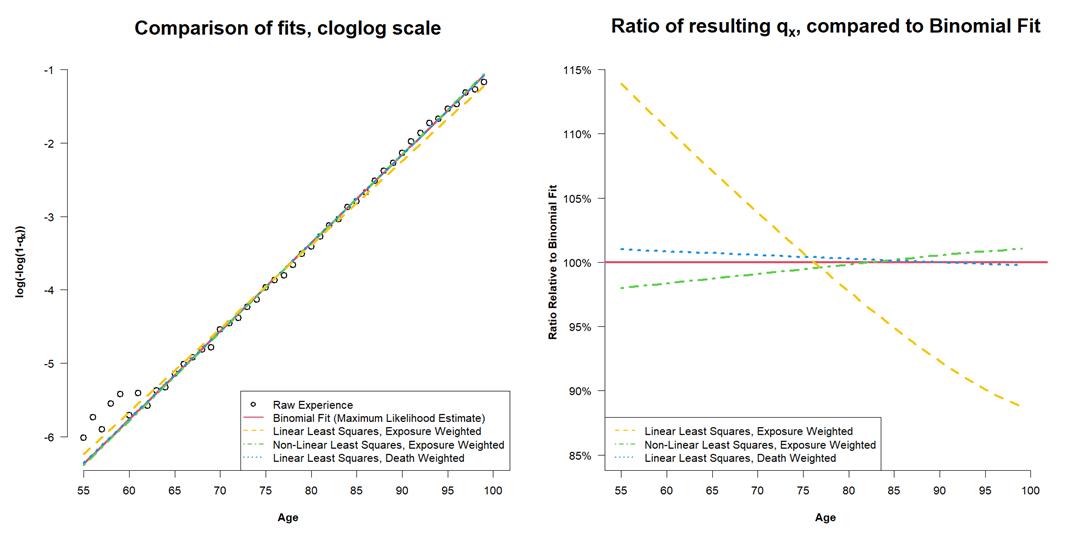
**1.**Maximum likelihood estimation (via a GLM), with the q\_x following a binomial distribution (the “Binomial Fit”)

**2.**Linear least squares with exposures as the weight è I believe this is consistent with your notes on function fitting

**3.**Non-linear least squares with exposures as the weight è this is consistent with how I believe you would normally apply Whittaker-Henderson, that is to the non-transformed q\_x with exposures as the weight

**4.**Linear least squares with deaths as the weight

The chart on the left provides the fitted lines (on a cloglog scale), while the one on the right plots the ratios of the q\_x’s to those found under the Binomial Fit.



Gompertz is not a great fit to this experience, but in the left chart 3 of the 4 lines end up being visually right on top of each other and more-or-less fit to the experience except at the youngest ages where there would be the most volatility. The one line that is inconsistent is the dashed yellow one, with corresponds to exposure-weighted linear least squares. The chart on the right renders the difference even clearer, with the ratios of the exposure-weighted linear least squares result to the binomial one being very different from the other approaches, both in relative and absolute terms. The death-weighted linear least squares appears to be the most consistent with the binomial maximum likelihood estimate, though the exposure-weighted non-linear least squares result is not too far off either.

What is happening here is that the best weights to use for weighted least squares are the reciprocal of the variances, and a transformation such as the cloglog one used here also changes the variance structure. For the non-transformed q\_x, it follows from the underlying binomial distribution that the variances at each age can be approximated by an expression proportional to [1 / exposure], so exposures end up working well enough as weights under non-linear least squares. However, under the cloglog transformation it can be shown (again following from the binomial distribution) that the variance can be approximated by an expression proportional to [1 / deaths], and so it then follows that deaths work best for weights under linear least squares since the cloglog transformation is employed there.

I should note that the differences between using exposures or deaths as the weights would generally be less extreme than what is seen above where the fit of the specific function is better. Since Gompertz is not a particularly good fit in this case, the difference is rather large. But it is still of course important to have appropriate weights in all situations.

I hope that makes it reasonably clear why deaths would make for a much better least squares weight than exposures when fitting to log-, logit-, or cloglog-transformed q\_x’s, but happy to provide further details if it would be helpful.

Regards,

Stefan

1. I mostly use the word “function”. Some prefer the words “curve”, “expression”, “equation”, or “model”. All are meant as synonyms. However, I particularly avoid “model” because it has a different meaning within professional standards. [↑](#footnote-ref-1)
2. Although the approximation is less accurate than one might like, my testing failed to find a better approximation that yielded an improvement in the accuracy of the resulting parameters. [↑](#footnote-ref-2)
3. This is the same dll that has the functions for Whittaker-Henderson graduation in one and two dimensions. Currently MortFn is available only in the 64-bit dll. Let me know if you need the 32-bit version. [↑](#footnote-ref-3)