RELAP - Revista Latinoamericana de Población
Vol. 18:e202315
http://doi.org/10.31406/relap2024.v18e202315
ISSN digital: 2393-6401

Modernizing Demographic Computations: Some Illustrative Examples

Modernizando los cómputos demográficos: algunos ejemplos ilustrativos

Ralph Hakkert
ralph_hakkert@hotmail.com
ORCID 0009-0005-1060-7265
Independent consultant

Matheus Menezes
matheus.menezes@hotmail.com.br
ORCID 0000-0002-8164-6182
Temporary professor of Demography at Cedeplar/UFMG

Abstract

Although the use of computers and sophisticated software, particularly R, has become common in demography, this has not led to a revision of some traditional demographic algorithms, which were originally designed to overcome computational obstacles that nowadays no longer exist. This article argues that, rather than implementing the existing algorithms in today's powerful software, it is time to take advantage of this software to modify the algorithms themselves. It illustrates this idea with three examples: Sprague interpolation, the Relational Gompertz method for fertility estimation, and the estimation of infant and child mortality from proportions of surviving children by mother's age. In each case, it is shown that the analysis can be made simpler, more direct, intuitive, transparent, and precise by making use of the resources offered by software such as R. It is also shown that the greater flexibility of these algorithms creates opportunities for extending each method.

Keywords

Demographic Algorithms
R Language
Sprague Method
Relational Gompertz Method
Infant and Child Mortality Estimation

Resumen

Aunque el uso de computadoras y software sofisticado, particularmente R, se ha vuelto común en la demografía, esto no ha llevado a una revisión de algunos algoritmos demográficos tradicionales, los cuales fueron originalmente diseñados para superar obstáculos computacionales que actualmente ya no

Palabras clave

Algoritmos demográficos
Lenguaje R

existen. Este artículo argumenta que, en lugar de implementar los algoritmos existentes en el poderoso software actual, es hora de aprovechar el software para modificar los propios algoritmos. Ilustra esta idea con tres ejemplos: la interpolación de Sprague, el método relacional de Gompertz para estimar la fecundidad y la estimación de la mortalidad infantil y en la niñez a partir de las proporciones de los hijos sobrevivientes por edad de la madre. En cada caso, se muestra que el análisis se puede hacer más simple, directo, intuitivo, transparente y preciso usando los recursos que ofrece un software como R. También se muestra que la mayor flexibilidad de estos algoritmos crea oportunidades para extender cada método.

Método de Sprague
Método Gompertz Relacional
Estimación de mortalidad infantil y en la niñez

Enviado: 19/04/23
Aceptado: 27/01/24

Introduction

This article is a manifest of sorts. Although it contains three proposals for demographic analysis, these are not primarily intended as actual methodological innovations but rather as examples to advocate a broader principle. They are not "software packages" as they could be refined and extended in various ways to make them more user-friendly and useful in actual research applications. Here, their purpose is merely to illustrate the general point that the current algorithms used to implement a limited but significant number of widely used demographic techniques are outdated, as they were designed to circumvent operational problems that no longer exist. These techniques include a few traditional data fitting and interpolation methods, which continue to be popular despite the availability of more recent alternatives. More importantly, however, many of the indirect estimation techniques that were developed in the 1960s and 1970s could benefit from an update of their numerical implementations. Although the use of computers and sophisticated software such as R or even more conventional tools such as Excel, which is still far superior to the software available in the 1960s and 1970s, has now become commonplace, this has not led to a retooling of these methods with more modern algorithms. Rather, we now often find ourselves in a contradictory situation where software is being developed to automate the execution of old algorithms that were designed precisely to overcome the lack of such resources in the past.

This situation is all the more incongruous given the pace of recent innovations, greatly aided by the availability of advanced computer software, in the border area between demography and statistics, such as probabilistic population projection methods (Raftery, Alkema & Gerland, 2014), Bayesian estimation methods for maternal mortality (Alkema et al., 2017) or age-specific migration (Alexander, Yeung & Riffe, 2022), and non-parametric graduation methods (Mazza & Punzo, 2011), to mention just a few. The stochastic generalization of Brass's P/F method by Baker, Alcantara and Ruan (2011) is an example of a recent innovation in the area of interest to this article, but their algorithm is based on a different approach.

The examples discussed in this article illustrate slightly different ways in which the implementation of some well-established methods seems to have become somewhat frozen in time. The conventional implementation of the Sprague interpolation method discussed in the first example, is usually convenient enough, but its generalization to non-standard cases is not widely known, even though it is quite easy to implement in R. The example of the Relational Gompertz method illustrates a somewhat more consequential issue, namely that the need for easy computability has led to a rather cumbersome implementation of a principle that is actually quite straightforward. Using the computational resources of R, this conceptual simplicity of the model can be restored, and some refinements and extensions can be introduced. The final example of infant and child mortality estimation follows the underlying logic of the simulation process by which the well-known and simple formulas described in Manual X (UN Population Division, 1983) and Moultrie et al. (2013) were generated but implements this process explicitly, something that was not possible in the times of Manual X. By doing so, the method becomes more transparent and more amenable to alternative assumptions, such as alternative life table models and the possibility of fertility and mortality variations in the recent past.

Rationale for the proposed approach

The main justifications for a retooling of some demographic methods, including indirect estimation methods, are the following.

Directness: The traditional implementations of the methods discussed in this paper – and others that are similar – typically need to make a separation between the underlying theoretical relationships, expressed by demographic models, and the tables, regression equations, and other practical tools needed to make the parameters of these models computable. The algorithms proposed in this paper allow a greater approximation between model parameters and the procedures for estimating them.

Intuitiveness/Transparency: Those of us who have taught indirect estimation methods know how difficult it is to keep the focus on the underlying rationale of these methods rather than becoming absorbed by their operational aspects, otherwise known as the “cookbook”. This is also true of the first example, Sprague interpolation, whose application is easy to explain but whose underlying logic often mystifies students. The algorithms proposed here are easier to explain and apply in the context of the theoretical relationships and models underlying them, provided that students are familiar with R or some advanced features of Excel.

Flexibility: The tables, regression equations, and other practical tools mentioned above need to be elaborated for each variant (e.g., each model life table family) separately. In practice, this constrains the number of variants that can be elaborated. The algorithms proposed here are much easier to adapt to alternative model assumptions.

Extendability: The flexibility of these methods also makes them easier to extend, not only in terms of alternative model variants, such as model life table families, but also in terms of additional parameters, e.g., to express changing fertility or mortality. In the traditional formulation of the methods, such extensions would also be theoretically possible, but their inclusion would make their execution operationally very cumbersome and confusing.

The algorithms suggested here are also more precise in that they better represent the theoretical models underlying the estimation methods. It is important, however, to note that by themselves, they do not constitute new estimation models and that the quality of the estimates derived from them is only as good as the underlying models allow. Perhaps the example of the Lotka equation (Dublin & Lotka, 1925) can illustrate this. The Lotka equation links the age-specific fertility *f(x)* and the survival function *p(x)* of a closed population to its intrinsic growth rate r through the following simple and well-known relationship:

$$\int_0^\infty e^{-rx} f(x)p(x)dx = 1$$

Because this expression does not have a simple analytical solution for *r*, Lotka proposed an approximate quadratic formula with coefficients based on *f(x)* and *p(x)*, which can be solved for *r*. Coale (1957) improved on this algorithm through a simple iterative procedure to approximate *r*, which is still in use today. Even this algorithm, however, has now become somewhat redundant because the *optim* and *optimize* procedures of R, like the *Solver* procedure of Excel, internalize the iteration process, thus making it possible to express r directly. The resulting solution is more precise than Lotka's and simpler and more direct than Coale's.

The point to be made here is that the greater precision of r neither changes stable population theory or its underlying assumptions nor does it guarantee that the estimates of any demographic parameters derived from r will be better than those based on Lotka's original approximation. The quality of these estimates depends much more on how well the stable population model represents the actual population dynamics than on the precision with which r is determined. Likewise, the algorithms suggested in this article better represent the underlying theoretical models of the estimation methods, in addition to the advantages mentioned above, but the quality of the resulting estimates still depends on the validity of the models themselves.

Sprague method

The generalization of Sprague interpolation to non-standard age intervals is probably the least consequential of the three examples provided here, but it does illustrate the point being made. Regardless of whether the Sprague method is the best way to interpolate or disaggregate population data, given the availability of several potentially better alternatives, such as the method proposed by Rizzi, Gampe and Eilers (2015), it is still one of the most used and least understood methods used for this purpose. Most demographers know it through the multiplier schedule on page 727 of Siegel and Swanson (2004), which disaggregates 5-year age groups into single ages, but few know how they were obtained or how to generalize them to situations where the intervals have unequal length or need to be split into non-standard parts. It does not help that the explanation of the method provided by Siegel and Swanson is stated in terms of difference formulas that require intervals of equal length.

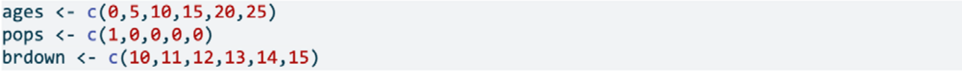
The Graduate_sprague function in DemoTools (Riffe et al., 2019), written in R, is entirely based on these multipliers. It does allow for intervals of unequal length but resorts to reclassifying these into standard 5-year age groups, using simple uniform distributions, before disaggregating the intervals into fifths with the standard multipliers. While the errors introduced this way may often be acceptable (see the example at the end of this section), the Sprague algorithm for disaggregating population distributed over five intervals of arbitrary length into smaller intervals of different sizes is actually quite simple. It involves fitting two 4th-degree polynomials to the cumulative population function, one for points 1-5 (P1-5) and one for points 2-6 (P2-6), and then interpolating the interval between points 3 and 4 by means of a 5th-degree polynomial that has the same value and first and second derivatives as P1-5 at point 3 and the same as P2-6 at point 4. The same logic can then be applied to successive intervals: 2-6 and 3-7, 3-7 and 4-8, etc. (Grupo de Foz, 2021, p. 673-675). The following program in R applies this routine, with the use of some basic matrix algebra available in R, to intervals 1-5 and 2-6:

```
# Data entry (example from the 2010 census of Brazil)
ages <- c(4,6,11,15,18,25) # Ages that delimit the 5 data intervals
pops <- c(2955845,7924037,6932267,5223476,11965621) # Populations in the age intervals
# If there are only four relevant intervals, the remaining one may be filled with a 0
brdown <- c(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25)
# Ages that delimit the interpolation intervals

# Algorithm
cumpop <- c(0,cumsum(pops)); result <- integer(length(brdown)-1)
# Definition of cumulative populations and initialization of the results vector
sprague <- function(x,y,intval) {
  # Computes the coefficients of the Sprague interpolation of x and y in interval intval
  x1 <- x[1:5]; x2 <- x[2:6]; y1 <- y[1:5]; y2 <- y[2:6]
  res1 <- solve(cbind(x1^4,x1^3,x1^2,x1,x1^0),y1)
  # res1 defines the coefficients of a 4th-degree polynomial to fit the first 5 data
points
  res2 <- solve(cbind(x2^4,x2^3,x2^2,x2,x2^0),y2)
  # Idem for the last five data points
  if (intval < 3) coefs <- c(0,res1)
  if (intval > 3) coefs <- c(0,res2)
  if (intval == 3) {
    mat3 <- matrix(c(x[3]^(5:0),5:0*x[3]^c(4:0,0),c(20,12,6,2,0,0)*x[3]^c(3:0,0,0),
                     x[4]^(5:0),5:0*x[4]^c(4:0,0),c(20,12,6,2,0,0)*x[4]^c(3:0,0,0)),
                     nrow=6,byrow=TRUE)
    vec3 <- c(y[3],sum(4:1*res1[1:4]*x[3]^(3:0)),sum(c(12,6,2)*res1[1:3]*x[3]^(2:0)),
              y[4],sum(4:1*res2[1:4]*x[4]^(3:0)),sum(c(12,6,2)*res2[1:3]*x[4]^(2:0)))
    coefs <- solve(mat3,vec3)
    # In the middle panel, a 5th degree polynomial has the same function value, 1st and
2nd
    # derivatives as the res1 polynomial on the left and the res2 polynomial on the right
  }
  coefs
}
curve <- function(x,y,point) { # Converts the coefficients into points on the curve
  sum(sprague(x,y,sum(point>=x))*point^(5:0))
}

# Presentation of result
for (i in 1:(length(brdown)-1)) result[i] <- curve(ages,compop,brdown[i+1])-
  curve(ages,compop,brdown[i])
result
 [1] 1469249 1486596 1512914 1545828 1582963 1621942 1660390 1698334 1733679
[10] 1750659 1749594 1746522 1742083 1734870 1726105 1717009 1708803 1702709
[19] 1699947 1701740 1709308
```

As a special case, consider the following specification of the data section:



```
ages <- c(0,5,10,15,20,25)
pops <- c(1,0,0,0,0)
brdown <- c(10,11,12,13,14,15)
```

This produces the vector (-0.0128, -0.0016, 0.0064, 0.0064, 0.0016), i.e., the first column of the set of multipliers presented in Siegel and Swanson (2004, p. 727) for splitting the middle panel. Similarly, the same specification, but with pops <- c(0,1,0,0,0), produces the second column of this table: (0.0848, 0.0144, -0.0336, -0.0416, -0.0240). The other columns and the multipliers for the other panels can be reproduced analogously.

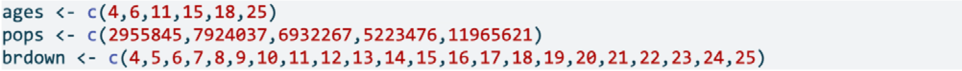
Admittedly, in demography the breakdown of 5-year age intervals into fifths is by far the most common use of Sprague interpolation, so the existing multipliers are usually sufficient. This is not necessarily true for other fields of application, such as income statistics. Yet, even in demography there are non-standard situations that can be addressed with the above program. Take, for example, the not uncommon case where data are available for age 0 and for the 1-4, 5-9, and 10-14 year age intervals. It may not be entirely without interest to know that in this case, using the above program, either the 1-4 year age interval can be split directly or the following multipliers can be calculated to interpolate the populations aged 1, 2, 3, and 4 years:

Table 1

Ages	Age 0	Ages 1-4	Ages 5-9	Ages 10-14
1	0.416	0.177905	-0.030095	0.004571
2	0.032	0.270095	-0.025905	0.003429
3	-0.184	0.292190	0.004190	-0.001143
4	-0.264	0.259810	0.051810	-0.006857

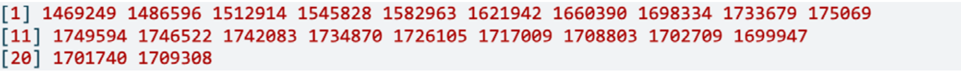
Unlike what happens in the standard case, which produces exact fractions, the last three columns had to be rounded to six digits, but this is merely a presentation issue.

As another application, consider the following specification, the one shown in the R script above:



```
ages <- c(4,6,11,15,18,25)
pops <- c(2955845,7924037,6932267,5223476,11965621)
brdown <- c(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25)
```

These data describe the population under age 25 of the 2010 Brazilian census according to school ages (4-5, 6-10, 11-14, 15-17, 18-24) rather than conventional 5-year age groups. To simplify the presentation and avoid multiple applications of the program, it is assumed that no age-specific information is available beyond age 25. The *brdown* vector specifies a request for a breakdown by single years, which generates the following result in R:



```
[1] 1469249 1486596 1512914 1545828 1582963 1621942 1660390 1698334 1733679 175069
[11] 1749594 1746522 1742083 1734870 1726105 1717009 1708803 1702709 1699947
[20] 1701740 1709308
```

Because vectors in R always start at position 1, vector positions must be increased by 3 to obtain the correct ages. Taking this into account, it is easily verified that the numbers by a single year of age sum exactly to the original age groupings.

Compare this to the procedure referred to earlier, in which the original age groups are first broken down into single ages using uniform distributions, rearranged into 5-year age groups, and finally disaggregated again by using the conventional multipliers. Unlike the results displayed above, the age distribution by single years in this case does not exactly reproduce the original data. For example, the reaggregated population for ages 11-14 will be *6,856,766* rather than *6,932,267*, and the reaggregated population for ages 6-10 will be *7,981,666* rather than *7,924,037*. For some applications, this minor transfer of population between age groups may not be a serious problem. However, it begs the question why one would do this if an exact solution is available.

The program can also be adapted to other formats, such as the breakdown of populations or events contained in squares within the Lexis diagram into their corresponding triangles, for the purpose of cohort analysis. Such extensions will not be shown here.

Relational Gompertz method

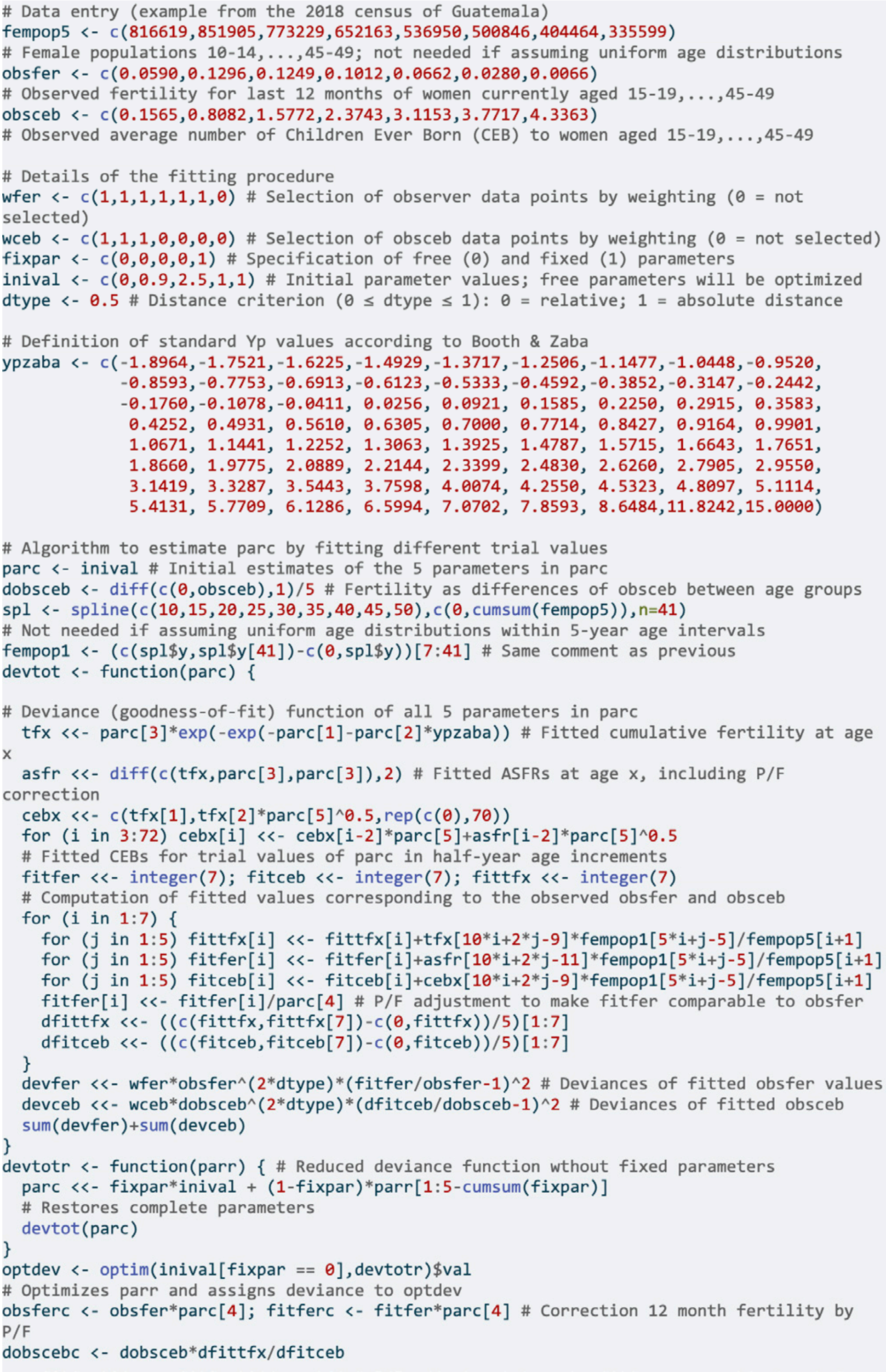
The proposition put forward in this article is not limited to the context of indirect estimation methods, nor does it question the validity of the demographic models underlying these methods. Yet, it is indirect estimation methods that could probably benefit most from a revision of the algorithms used in their implementation.

The one-census Relational Gompertz method for fertility estimation has become one of the most-used methods for this purpose nowadays, largely replacing the older Brass P/F methods. Its main advantage is that it considers a wider range of comparisons of current fertility F (last 12 months) with life-long fertility P (Children Ever Born) than the traditional comparisons of P2 with F2 or P3 with F3. Although the one-census variant of the method is less sensitive to fertility changes than the P/F methods, it does not incorporate fertility change explicitly; the latter can be done using the two-census Synthetic Relational Gompertz variant. In its essence, the method comes down to a rather simple adjustment of a theoretical curve, the Relational Gompertz curve, to a set of 7 current fertility (F) data points and 7 data points describing the average number of Children Ever Born by mothers' age group (P). The parameters that need to be estimated are the location parameter α and the dispersion parameter β of the Relational Gompertz curve, the overall fertility level TFR, and the adjustment factor P/F that needs to be applied to the current fertility levels, which should also be shifted half a year down.

In the late 1970s and early 1980s, when the method was first proposed, finding the optimum levels of these parameters to minimize the distance between observed and fitted values on a non-linear curve was a rather daunting task, especially for demographers in developing countries, who had limited access to mathematical optimization algorithms. To circumvent this obstacle, Zaba (1981) devised a rather ingenious albeit cumbersome method to linearize the problem so that it would be amenable to more widely known fitting procedures such as linear regression. In order to do so, however, it was necessary to introduce several auxiliary functions z(i), e(i), and g(i), which were not intuitive and made the procedure rather opaque.

Nowadays, the method continues to be applied in much the same way as in the early 1980s, and it has recently been programmed in R, in the R/gomp_pf.R program (Monteiro da Silva, s.d.), even though the problem that motivated this particular implementation has long ceased to exist. Even in Excel, but certainly in R, the once formidable challenge of optimizing four parameters to obtain the best fit of data points to a non-linear function has now become rather trivial and can be met head-on. The main advantage of doing so is conceptual clarity: rather than requiring several unintuitive auxiliary functions, the problem can now be stated in terms of basic concepts that every demographer is familiar with. A secondary advantage is that this more straightforward formulation of the problem is more precise, in the sense that was explained earlier, and, more importantly, extendable.

The following program in R estimates the four relevant parameters – and optionally a fifth one – directly by fitting the observed values to the theoretical Relational Gompertz curve describing the Age Specific Fertility Rates:



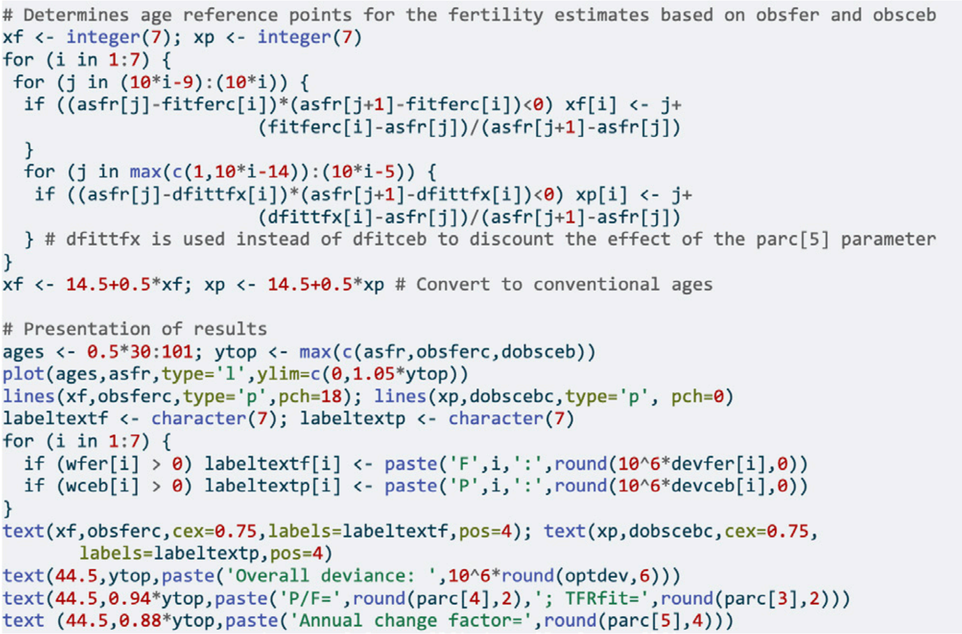
```
# Data entry (example from the 2018 census of Guatemala)
fempop5 <- c(816619,851905,773229,652163,536950,500846,404464,335599)
# Female populations 10-14,...,45-49; not needed if assuming uniform age distributions
obsfer <- c(0.0590,0.1296,0.1249,0.1012,0.0662,0.0280,0.0066)
# Observed fertility for last 12 months of women currently aged 15-19,...,45-49
obsceb <- c(0.1565,0.8082,1.5772,2.3743,3.1153,3.7717,4.3363)
# Observed average number of Children Ever Born (CEB) to women aged 15-19,...,45-49

# Details of the fitting procedure
wfer <- c(1,1,1,1,1,1,0) # Selection of observer data points by weighting (0 = not
selected)
wceb <- c(1,1,1,0,0,0,0) # Selection of obsceb data points by weighting (0 = not selected)
fixpar <- c(0,0,0,0,1) # Specification of free (0) and fixed (1) parameters
inival <- c(0,0.9,2.5,1,1) # Initial parameter values; free parameters will be optimized
dtype <- 0.5 # Distance criterion (0 ≤ dtype ≤ 1): 0 = relative; 1 = absolute distance

# Definition of standard Yp values according to Booth & Zaba
ypzaba <- c(-1.8964,-1.7521,-1.6225,-1.4929,-1.3717,-1.2506,-1.1477,-1.0448,-0.9520,
            -0.8593,-0.7753,-0.6913,-0.6123,-0.5333,-0.4592,-0.3852,-0.3147,-0.2442,
            -0.1760,-0.1078,-0.0411, 0.0256, 0.0921, 0.1585, 0.2250, 0.2915, 0.3583,
             0.4252, 0.4931, 0.5610, 0.6305, 0.7000, 0.7714, 0.8427, 0.9164, 0.9901,
             1.0671, 1.1441, 1.2252, 1.3063, 1.3925, 1.4787, 1.5715, 1.6643, 1.7651,
             1.8660, 1.9775, 2.0889, 2.2144, 2.3399, 2.4830, 2.6260, 2.7905, 2.9550,
             3.1419, 3.3287, 3.5443, 3.7598, 4.0074, 4.2550, 4.5323, 4.8097, 5.1114,
             5.4131, 5.7709, 6.1286, 6.5994, 7.0702, 7.8593, 8.6484,11.8242,15.0000)

# Algorithm to estimate parc by fitting different trial values
parc <- inival # Initial estimates of the 5 parameters in parc
dobsceb <- diff(c(0,obsceb),1)/5 # Fertility as differences of obsceb between age groups
spl <- spline(c(10,15,20,25,30,35,40,45,50),c(0,cumsum(fempop5)),n=41)
# Not needed if assuming uniform age distributions within 5-year age intervals
fempop1 <- (c(spl$y,spl$y[41])-c(0,spl$y))[7:41] # Same comment as previous
devtot <- function(parc) {

# Deviance (goodness-of-fit) function of all 5 parameters in parc
  tfx <<- parc[3]*exp(-exp(-parc[1]-parc[2]*ypzaba)) # Fitted cumulative fertility at age
x
  asfr <<- diff(c(tfx,parc[3],parc[3]),2) # Fitted ASFRs at age x, including P/F
correction
  cebx <<- c(tfx[1],tfx[2]*parc[5]^0.5,rep(c(0),70))
  for (i in 3:72) cebx[i] <<- cebx[i-2]*parc[5]+asfr[i-2]*parc[5]^0.5
  # Fitted CEBs for trial values of parc in half-year age increments
  fitfer <<- integer(7); fitceb <<- integer(7); fittfx <<- integer(7)
  # Computation of fitted values corresponding to the observed obsfer and obsceb
  for (i in 1:7) {
    for (j in 1:5) fittfx[i] <<- fittfx[i]+tfx[10*i+2*j-9]*fempop1[5*i+j-5]/fempop5[i+1]
    for (j in 1:5) fitfer[i] <<- fitfer[i]+asfr[10*i+2*j-11]*fempop1[5*i+j-5]/fempop5[i+1]
    for (j in 1:5) fitceb[i] <<- fitceb[i]+cebx[10*i+2*j-9]*fempop1[5*i+j-5]/fempop5[i+1]
    fitfer[i] <<- fitfer[i]/parc[4] # P/F adjustment to make fitfer comparable to obsfer
    dfittfx <<- ((c(fittfx,fittfx[7])-c(0,fittfx))/5)[1:7]
    dfitceb <<- ((c(fitceb,fitceb[7])-c(0,fitceb))/5)[1:7]
  }
  devfer <<- wfer*obsfer^(2*dtype)*(fitfer/obsfer-1)^2 # Deviances of fitted obsfer values
  devceb <<- wceb*dobsceb^(2*dtype)*(dfitceb/dobsceb-1)^2 # Deviances of fitted obsceb
  sum(devfer)+sum(devceb)
}
devtotr <- function(parr) { # Reduced deviance function wthout fixed parameters
  parc <<- fixpar*inival + (1-fixpar)*parr[1:5-cumsum(fixpar)]
  # Restores complete parameters
  devtot(parc)
}
optdev <- optim(inival[fixpar == 0],devtotr)$val
# Optimizes parr and assigns deviance to optdev
obsferc <- obsfer*parc[4]; fitferc <- fitfer*parc[4] # Correction 12 month fertility by
P/F
dobscebc <- dobsceb*dfittfx/dfitceb
```



```
# Determines age reference points for the fertility estimates based on obsfer and obsceb
xf <- integer(7); xp <- integer(7)
for (i in 1:7) {
 for (j in (10*i-9):(10*i)) {
  if ((asfr[j]-fitferc[i])*(asfr[j+1]-fitferc[i])<0) xf[i] <- j+
                        (fitferc[i]-asfr[j])/(asfr[j+1]-asfr[j])
  }
  for (j in max(c(1,10*i-14)):(10*i-5)) {
   if ((asfr[j]-dfittfx[i])*(asfr[j+1]-dfittfx[i])<0) xp[i] <- j+
                        (dfittfx[i]-asfr[j])/(asfr[j+1]-asfr[j])
  } # dfittfx is used instead of dfitceb to discount the effect of the parc[5] parameter
}
xf <- 14.5+0.5*xf; xp <- 14.5+0.5*xp # Convert to conventional ages

# Presentation of results
ages <- 0.5*30:101; ytop <- max(c(asfr,obsferc,dobsceb))
plot(ages,asfr,type='l',ylim=c(0,1.05*ytop))
lines(xf,obsferc,type='p',pch=18); lines(xp,dobscebc,type='p', pch=0)
labeltextf <- character(7); labeltextp <- character(7)
for (i in 1:7) {
  if (wfer[i] > 0) labeltextf[i] <- paste('F',i,':',round(10^6*devfer[i],0))
  if (wceb[i] > 0) labeltextp[i] <- paste('P',i,':',round(10^6*devceb[i],0))
}
text(xf,obsferc,cex=0.75,labels=labeltextf,pos=4); text(xp,dobscebc,cex=0.75,
       labels=labeltextp,pos=4)
text(44.5,ytop,paste('Overall deviance: ',10^6*round(optdev,6)))
text(44.5,0.94*ytop,paste('P/F=',round(parc[4],2),'; TFRfit=',round(parc[3],2)))
text (44.5,0.88*ytop,paste('Annual change factor=',round(parc[5],4)))
```

The program owes some of its length to the fact that it contains several refinements and extensions of the standard specification. A bare bones version, without these additions, could be cut by about half. However, even with these additions, the length of the program, without comments, is less than a tenth of that of the similarly uncommented program R/gomp_pf.R.

In addition to the comments included in the program itself, the following details need to be clarified:

1. The main objective of the program is to optimize the *parc* vector through the *optim* procedure of R (Nelder-Mead method),[1] in order to minimize the distance between the observed and fitted values of fertility during the past 12 months (*obsfer* and *fitfer*) and Children Ever Born (*obsceb* and *fitceb*). The graph generated by the program includes the values of *devfer* (goodness-of-fit of individual F points), *devceb* (goodness-of-fit of individual P points), *optdev* (overall deviance or goodness-of-fit), *parc[4]* and *parc[5]* (see below). Optionally, the user can request the values of other vectors, such as the complete

---

[1] The Nelder-Mead method is not appropriate when there is only one free parameter, but this is unlikely to occur in this particular application. It may occur in the case of the third example, of mortality estimation, where a slightly more robust procedure will be used.

parc vector, *fitfer*, *fitceb*, xf or xp (see below) or the complete set of fitted ASFRs in increments of 0.5 years stored in *asfr*.

2. The *parc* vector contains the complete parameter set, consisting of

parc[1]:	The location parameter α of the Relational Gompertz specification;
parc[2]:	The dispersion parameter β of the Relational Gompertz specification;
parc[3]:	The overall fertility level TFR of the fitted curve;
parc[4]:	The P/F adjustment factor to correct the time reference error of the F series; and
parc[5]:	A factor expressing the annual change of fertility as one goes back in time. For example, a factor 1.01 implies that fertility in all age groups was successively 1% higher for each year in the past. Note that this will not greatly affect fitted values of Children Ever Born (CEB) in the youngest age groups, but the effect increases with age. In the standard application of the model, this parameter is fixed at a value of 1, but other values can be tried, particularly if fertility decline is known to have occurred in the past. Including *parc[5]* as a free parameter can lead to spurious results and should be done judiciously.

The *parr* vector is the reduced form of *parc* containing only the free parameters, in case the user chooses to fix some of them. If no parameters are fixed, *parr* is equal to *parc*, and the wrapper function *devtotr* is identical to *devtot*.

3. Unlike the *fempop5*, *obsfer*, and *avceb* vectors, which contain the data for particular applications, the Y_p vector usually does not vary. It contains the gompits of the standard fertility schedule defined by Booth and Zaba, as used by Moultrie et al. (2013). It could be read from an external file, but here, it has been included in the program for easy reference and to allow direct modification of the standard, should the user want to do this.

4. The refinements and extensions referred to above concern the following issues:

- Addition of a parameter *parc[5]* to describe fertility change over time, as explained under 2).
- The option to fix some of the parameters - usually *par[5]*, but sometimes *par[2]*, *par[3]*, or *par[4]*.
- The option to choose different kinds of optimality criteria (absolute square distance, proportional square distance, and everything in between), rather than depending on a somewhat untransparent error structure implied by the regression of the transformed values of *obsfer* and *obsceb*.
- Greater attention to the heterogeneity of the population age structure within each age interval and the choice of the appropriate age reference points for the F and P series. It is usually assumed that F refers to ages 17, 22, 27, etc., and P to ages 15, 20, 25, etc., but this can be somewhat misleading. In some age groups, it can cause the data points to appear different from the expected fitted values, even though they constitute a perfect fit. The penultimate section of the program computes the vectors xf and xp, which contain the ages at which the fitted values coincide exactly with the fertility curve.

5. The program produces a graph of fitted current fertility with the fitted data points. This graph reflects the P/F correction for the fertility of the past 12 months and corrects the average number of Children Ever Born for time trends, so that the displayed results reflect current levels, rather than referring to different time references in the past.

Even though the data for Guatemala used in the program (*obsfer* and *obsceb*) are merely an example, it is worth looking briefly at the results obtained and how they compare with conventional estimates. Application of the standard Relational Gompertz method, as described in Moultrie et al. (2013), yields a good fit when only F1-F6 and P1-P3 are considered. Under these conditions, a Total Fertility Rate (TFR) of 3.05 is obtained. Figure 1 replicates this analysis using the R script shown above. The settings for the fitting procedure are those shown in the second section of the R script. It sets the annual change factor *parc[5]* to 1, as in the standard method, and also limits the analysis to F1-F6 and P1-P3. The resulting TFR estimate of 3.03 is essentially identical to that of the standard algorithm.

Figure 1. Application of the restated Relational Gompertz method with *parc[5]*=1 and nine data points.

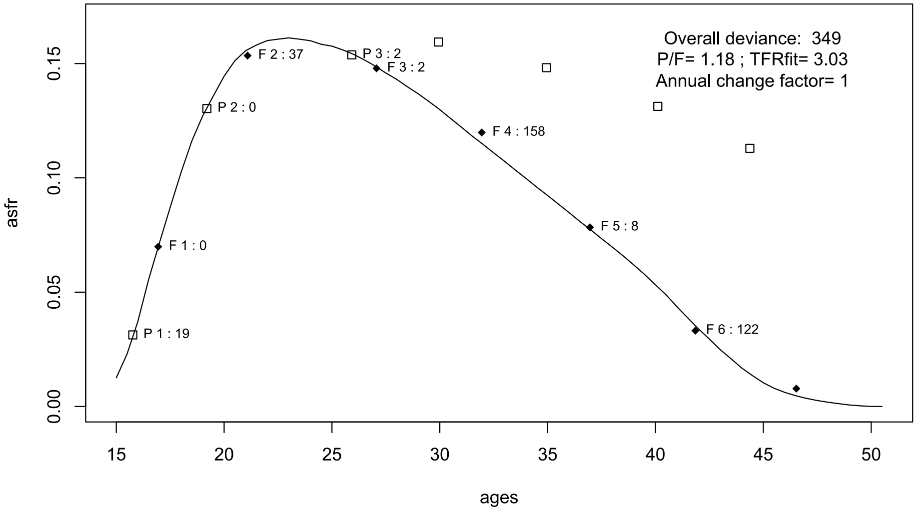
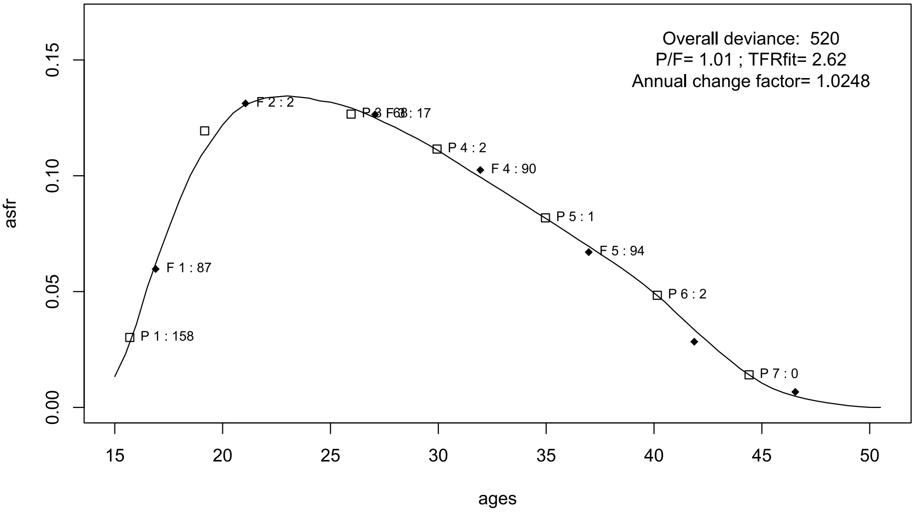


Figure 1 also shows how the P data points at the highest ages progressively move away from the corresponding F data points. This suggests that they reflect higher fertility in the past. Figure 2 repeats the analysis, this time with a free annual change factor *parc[5]*. Clearly, this is a much better fit. To improve it slightly more, the P2, F6, and F7 data points have been removed, but actually, this is hardly necessary. The TFR, in this case, becomes 2.62, fairly close to the 2018 estimate of 2.74 that the UN Population Division (2022) arrived at in its 2022 revision of World Population Prospects. Also note that the annual change factor is estimated at 2.48 % per year, which is very close to the 2.42 % average annual decline of the TFR between 1988 and 2018, according to the same source.

Although using a free annual change factor works quite well in this case, this must be done with some care because it can also produce spurious results. If a reasonable estimate of the rhythm of recent fertility decline is available, it is usually safer to enter this estimate into the program as a fixed parameter rather than letting the program determine the optimal value. The extension of the method to the two-census case, i.e., the Synthetic Relational Gompertz method, is not contemplated here, although the approach outlined above is well-suited to this task. In this variant, the parameter *parc[5]* could be removed. However, if a previous census can be used to obtain a viable estimate of *parc[5]*, that is usually a viable alternative.

Figure 2. Application of the restated Relational Gompertz method with *parc[5]* free and eleven data points.



Estimation of infant and child mortality

Unlike the Relational Gompertz method in the previous section, which is burdened by unnecessarily complicated conventional computational procedures, the conventional methods for estimating infant and child mortality are deceptively simple. Both Manual X (UN Population Division, 1983) and its more recent update in Moultrie et al. (2013) provide straightforward formulas whereby the proportion of children of women of certain age groups that have died can be converted into probabilities of death between birth and specified exact ages, given a model life table family and one or more measures of the shape of the fertility curve. A set of multipliers is provided to that end. A similarly easy-to-apply procedure provides the time references for these estimates.

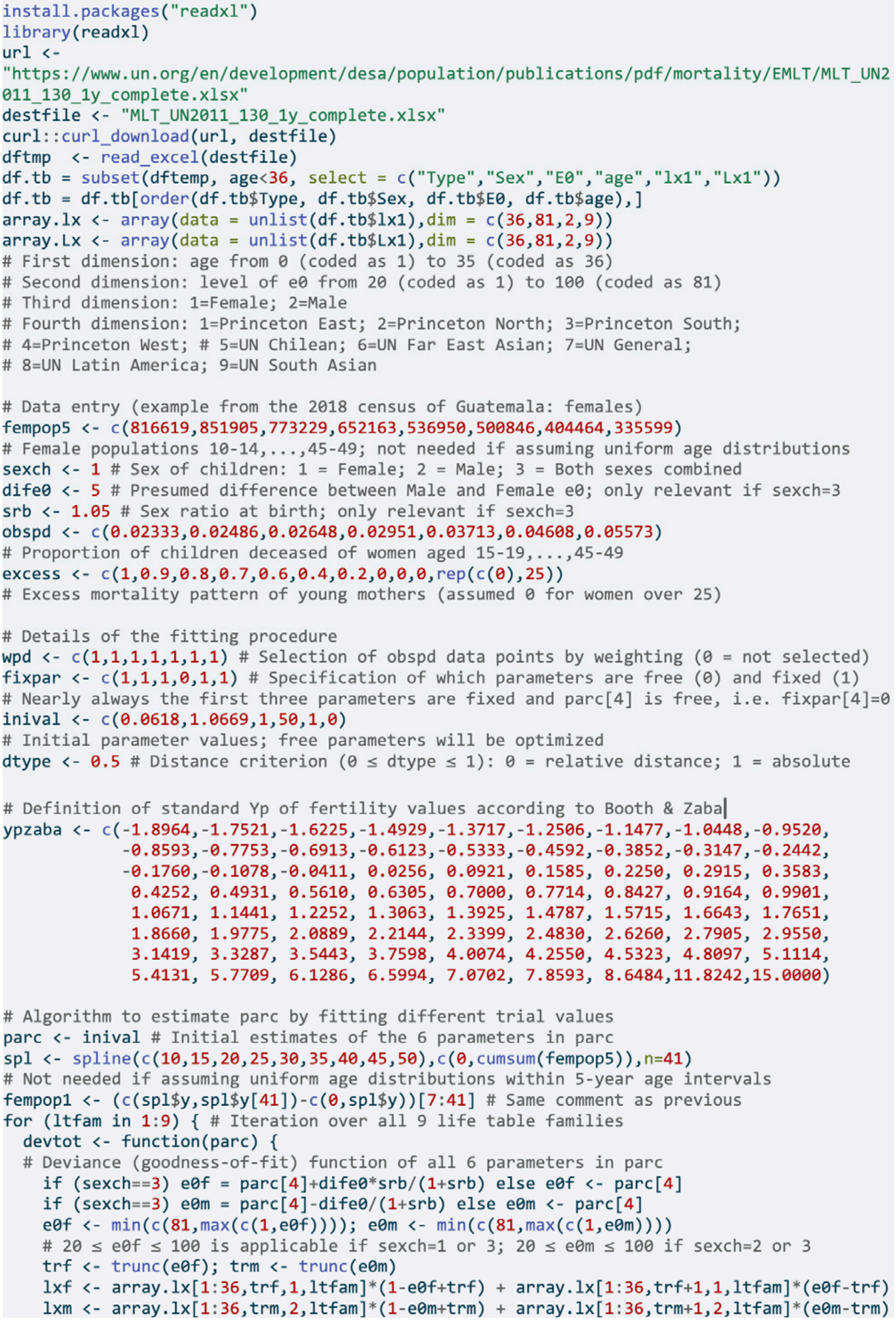
The limitation of these procedures is not that they are complicated but that they are based on simulations that are not transparent to the user and that require some fairly significant assumptions. Some of these are inevitable, such as the assumption that there is no correlation between the survival of mothers and children. In theory, it would be possible to incorporate such a correlation into the procedure presented below, but it would require some previous knowledge about the strength of this correlation, information that is usually difficult to come by. Other assumptions have to do with the fact that, in the process of converting the results of simulation exercises on which they are based into easy-to-apply multipliers, certain simplifying assumptions had to be made in order to keep the procedures manageable. The most important of these are the following:

1. As stated in Manual X (1983:74), " this method of estimation is based on the assumption that fertility and childhood mortality have remained constant in the recent past. If, for example, fertility has been changing, the ratios of average parities obtained from a cross-sectional survey will not replicate accurately the experience of any cohort of women and will not provide a good index of the distribution in time of the births to the women of each age group".

2. Infant and child mortality does not vary by the age of the mother.

As noted above, the methods require mortality to be stationary, but the mortality estimates associated with each age interval are fairly robust with respect to linear changes in mortality levels, provided that they are allocated to the correct time references. However, without knowing the rhythm of mortality change, estimating current mortality may be more challenging. One may select a particular life table indicator (usually q(1) or q(5)), use a model life table to convert all the mortality estimates for different points in time to this format, and then fit a time trend, but the quality of the fitted trend lines may vary, and the results may depend on the particular indicator chosen. It would be convenient to have a more comprehensive criterion for assessing the overall goodness-of-fit of a particular choice of model.

It should be noted that the program listing below, which makes use of the Princeton and United Nations set of 9 model life table families, does so for the sake of comparability, as most current methods use these models. However, the program could be adapted relatively easily to work with alternative specifications, such as Brass's logit model (Brass, 1971) or the more recent model proposed by Wilmoth et al. (2012), which are based on two parameters. The choice of the conventional Princeton and United Nations life table families poses an initial challenge, namely how to download this entire set of model life tables into the program. In the past, this alone would have been a major challenge as this information was not available in machine-readable form and not disaggregated by single years of age. However, thanks to the UN Population Division's website https://www.un.org/development/desa/pd/data/model-life-tables, downloading this entire set of model life tables has now become quite easy. The following program carries out the download, selects the relevant age groups (0-34 years), and limits the life table functions to ℓ_x and ${}_1L_x$, as the other life table functions will not be used.



```
install.packages("readxl")
library(readxl)
url <-
"https://www.un.org/en/development/desa/population/publications/pdf/mortality/EMLT/MLT_UN2
011_130_1y_complete.xlsx"
destfile <- "MLT_UN2011_130_1y_complete.xlsx"
curl::curl_download(url, destfile)
dftmp  <- read_excel(destfile)
df.tb = subset(dftemp, age<36, select = c("Type","Sex","E0","age","lx1","Lx1"))
df.tb = df.tb[order(df.tb$Type, df.tb$Sex, df.tb$E0, df.tb$age),]
array.lx <- array(data = unlist(df.tb$lx1),dim = c(36,81,2,9))
array.Lx <- array(data = unlist(df.tb$Lx1),dim = c(36,81,2,9))
# First dimension: age from 0 (coded as 1) to 35 (coded as 36)
# Second dimension: level of e0 from 20 (coded as 1) to 100 (coded as 81)
# Third dimension: 1=Female; 2=Male
# Fourth dimension: 1=Princeton East; 2=Princeton North; 3=Princeton South;
# 4=Princeton West; # 5=UN Chilean; 6=UN Far East Asian; 7=UN General;
# 8=UN Latin America; 9=UN South Asian

# Data entry (example from the 2018 census of Guatemala: females)
fempop5 <- c(816619,851905,773229,652163,536950,500846,404464,335599)
# Female populations 10-14,...,45-49; not needed if assuming uniform age distributions
sexch <- 1 # Sex of children: 1 = Female; 2 = Male; 3 = Both sexes combined
dife0 <- 5 # Presumed difference between Male and Female e0; only relevant if sexch=3
srb <- 1.05 # Sex ratio at birth; only relevant if sexch=3
obspd <- c(0.02333,0.02486,0.02648,0.02951,0.03713,0.04608,0.05573)
# Proportion of children deceased of women aged 15-19,...,45-49
excess <- c(1,0.9,0.8,0.7,0.6,0.4,0.2,0,0,0,rep(c(0),25))
# Excess mortality pattern of young mothers (assumed 0 for women over 25)

# Details of the fitting procedure
wpd <- c(1,1,1,1,1,1,1) # Selection of obspd data points by weighting (0 = not selected)
fixpar <- c(1,1,1,0,1,1) # Specification of which parameters are free (0) and fixed (1)
# Nearly always the first three parameters are fixed and parc[4] is free, i.e. fixpar[4]=0
inival <- c(0.0618,1.0669,1,50,1,0)
# Initial parameter values; free parameters will be optimized
dtype <- 0.5 # Distance criterion (0 ≤ dtype ≤ 1): 0 = relative distance; 1 = absolute

# Definition of standard Yp of fertility values according to Booth & Zaba
ypzaba <- c(-1.8964,-1.7521,-1.6225,-1.4929,-1.3717,-1.2506,-1.1477,-1.0448,-0.9520,
            -0.8593,-0.7753,-0.6913,-0.6123,-0.5333,-0.4592,-0.3852,-0.3147,-0.2442,
            -0.1760,-0.1078,-0.0411, 0.0256, 0.0921, 0.1585, 0.2250, 0.2915, 0.3583,
             0.4252, 0.4931, 0.5610, 0.6305, 0.7000, 0.7714, 0.8427, 0.9164, 0.9901,
             1.0671, 1.1441, 1.2252, 1.3063, 1.3925, 1.4787, 1.5715, 1.6643, 1.7651,
             1.8660, 1.9775, 2.0889, 2.2144, 2.3399, 2.4830, 2.6260, 2.7905, 2.9550,
             3.1419, 3.3287, 3.5443, 3.7598, 4.0074, 4.2550, 4.5323, 4.8097, 5.1114,
             5.4131, 5.7709, 6.1286, 6.5994, 7.0702, 7.8593, 8.6484,11.8242,15.0000)

# Algorithm to estimate parc by fitting different trial values
parc <- inival # Initial estimates of the 6 parameters in parc
spl <- spline(c(10,15,20,25,30,35,40,45,50),c(0,cumsum(fempop5)),n=41)
# Not needed if assuming uniform age distributions within 5-year age intervals
fempop1 <- (c(spl$y,spl$y[41])-c(0,spl$y))[7:41] # Same comment as previous
for (ltfam in 1:9) { # Iteration over all 9 life table families
  devtot <- function(parc) {
  # Deviance (goodness-of-fit) function of all 6 parameters in parc
    if (sexch==3) e0f = parc[4]+dife0*srb/(1+srb) else e0f <- parc[4]
    if (sexch==3) e0m = parc[4]-dife0/(1+srb) else e0m <- parc[4]
    e0f <- min(c(81,max(c(1,e0f)))); e0m <- min(c(81,max(c(1,e0m))))
    # 20 ≤ e0f ≤ 100 is applicable if sexch=1 or 3; 20 ≤ e0m ≤ 100 if sexch=2 or 3
    trf <- trunc(e0f); trm <- trunc(e0m)
    lxf <- array.lx[1:36,trf,1,ltfam]*(1-e0f+trf) + array.lx[1:36,trf+1,1,ltfam]*(e0f-trf)
    lxm <- array.lx[1:36,trm,2,ltfam]*(1-e0m+trm) + array.lx[1:36,trm+1,2,ltfam]*(e0m-trm)
```

```
    LLxf <- array.Lx[1:36,trf,1,ltfam]*(1-e0f+trf) +
array.Lx[1:36,trf+1,1,ltfam]*(e0f-trf)
    LLxm <- array.Lx[1:36,trm,2,ltfam]*(1-e0m+trm) +
array.Lx[1:36,trm+1,2,ltfam]*(e0m-trm)
    if (sexch==1) { # Fitted lx and LLx values for whole or fractional e0s
      lx <<- lxf/100000; LLx <<- LLxf/100000
    }
    if (sexch==2) {
      lx <<- lxm/100000; LLx <<- LLxm/100000
    }
    if (sexch==3) {
      lx <<- (lxf+sbr*lxm)/(100000*(1+sbr))
      LLx <<- (LLxf+sbr*LLxm)/(100000*(1+sbr))
    }
    # This completes the determination of the fitted life table for children
    tfx <<- exp(-exp(-parc[1]-parc[2]*ypzaba)) # Fitted cumulative fertility at age x
    asfr <<- diff(c(tfx,1,1),2) # Fitted ASFRs in half year age increments
    cebx <<- c(tfx[1],tfx[2]*parc[3]^0.5,rep(c(0),70))
    for (i in 3:72) cebx[i] <<- cebx[i-2]*parc[3]+asfr[i-2]*parc[3]^0.5
    # Fitted CEBs for trial values of parc in half-year age increments
    chdx <<- integer(35) # Initializes chdx, analogous to cebx for children that died
    lxcoh <<- rep(c(1),36); LLxcoh <<- rep(c(LLx[1]^(parc[5]^0.5)),35)
    for (i in 2:36) LLxcoh[i] <<- LLxcoh[i-1]^parc[5]*(LLx[i]/LLx[i-1])^(parc[5]^0.5)
    for (i in 2:36) lxcoh[i] <<- lxcoh[i-1]^parc[5]*(lx[i]/lx[i-1])^(parc[5]^0.5)
    # lx and LLx for cohorts, as cohorts pass through different mortality levels
    for (i in 1:35) chdx[i] <<- tfx[1]*parc[3]^i*(1-lxcoh[i+1]^(1+parc[6]*excess[1]))
    for (i in 1:35) {
      for (j in 1:i) chdx[i] <- chdx[i]+asfr[2*j-1]*parc[3]^(i-j+0.5)*
                                (1-LLxcoh[i-j+1]^(1+parc[6]*excess[j]))
    }
    fitceb <<- integer(7); fitpd <<- integer(7) # Computing fitted equivalents for obspd
    for (i in 1:7) {
      for (j in 1:5) fitceb[i] <<- fitceb[i] +
                                cebx[10*i+2*j-9]*fempop1[5*i+j-5]/fempop5[i+1]
      for (j in 1:5) fitpd[i] <<- fitpd[i] +
                                chdx[5*i+j-5]*fempop1[5*i+j-5]/fempop5[i+1]
    }
    fitpd <<- fitpd / fitceb
    devpd <<- wpd*obspd^(2*dtype)*(fitpd/obspd-1)^2 # Deviances of fitted obspd values
    sum(devpd)
  }
  devtotr <- function(parr) { # Reduced deviance function without fixed parameters
    parc <<- fixpar*inival + (1-fixpar)*parr[1:6-cumsum(fixpar)]
    # Restores complete parameters
    devtot(parc)
  }
  if (sum(fixpar)<5) optdev <- optim(inival[fixpar == 0],devtotr)$val else
    optdev <- optim(inival[fixpar == 0],devtotr,method='Brent',lower=1,upper=100)$val
  # Optimizes parr using Nelder-Mead or Brent-Dekker method and assigns deviance to optdev

  # Presentation of results
  cat(c('Life Table Model',ltfam,'; e0 =',round(parc[4]+19,2),'; Deviance =',
        1000000*round(optdev,6),'; fitpd =',round(fitpd,4)),'\n')
}
```

Again, the program is longer than strictly necessary because it contains some options, such as carrying out the analysis using a male life table, a female life table, or a life table for both sexes combined. Because it uses some of the same resources as the previous example, there is a fair amount of overlap. From a user viewpoint, the best strategy would actually be to merge both programs, but for presentation purposes, they are reproduced separately. However, the connection between both programs is evident from the fact that the mortality program uses the *parc[1]*, *parc[2]*, and *parc[5]* (here named *parc[3]*) parameters resulting from the fertility analysis, rather than using *obsceb[1]*, *obsceb[2]* and *obsceb[3]*, as in the conventional method.

As in the previous example, the *parc* and *parr* variables refer to the complete and reduced parameter sets, but the parameters included in *parc*, in this case, are somewhat different, namely:

parc[1]: The location parameter α of the Relational Gompertz specification (fertility);

parc[2]: The dispersion parameter β of the Relational Gompertz specification (fertility);

parc[3]: A factor expressing the annual change of fertility, i.e., the *parc[5]* parameter of the fertility model;

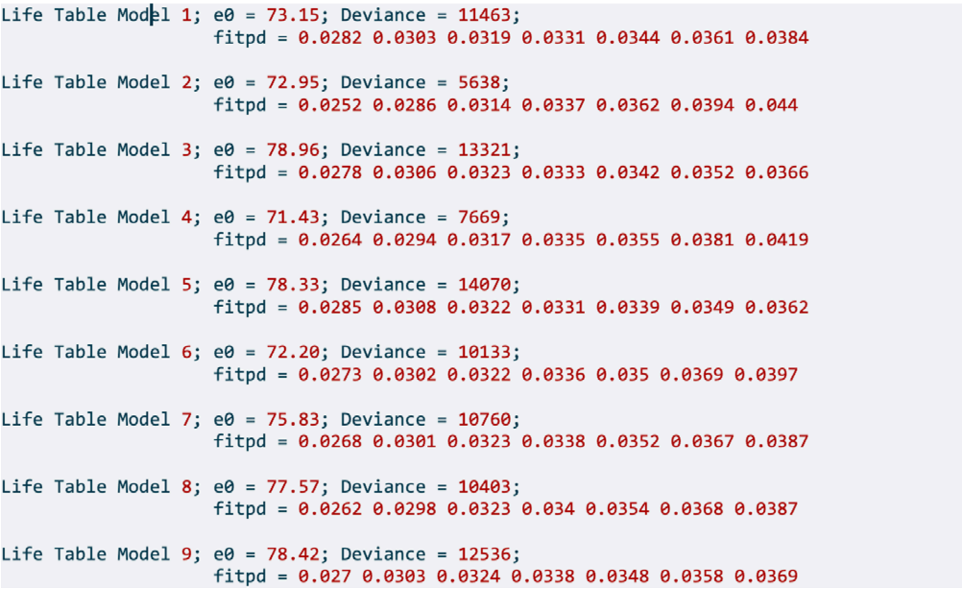
parc[4]: The level of mortality or life expectancy at birth;

parc[5]: A factor expressing the annual change of mortality; and

parc[6]: Level of excess mortality of children of young mothers. Note that the inclusion of this parameter filters out the excess mortality, which, therefore is excluded from the estimate of *par[4]*. Thus, strictly speaking, and using the definition of *excess* shown in the R script above, the estimate of *par[4]* only refers to children born to mothers over age 21. This is not unlike what happens in the conventional application of the method when the 15-19 age group is excluded. However, because some portion of children is born to mothers in this age group, both estimates contain a downward mortality bias. The advantage of including *par[6]* is that, although this is not shown here, it opens the possibility of explicitly computing the size of this bias.

Most of these parameters are fixed. Only *parc[4]* and *parc[6]* are usually free. More rarely, *parc[3]* and *parc[5]* can also be optimized. Missing here is the parameter that determines the life table family used. Because this is a nominal level variable, it is treated differently. The program implements all 9 alternatives, with their respective deviances, so that the user can decide which one provides the best fit. Finally, the user can request to see other indicators, such as the estimated complete parameter set *parc* or the deviances *devpd* for each age interval.

Again, a brief discussion of the results of the example is called for, to show that it yields results similar to the conventional algorithm, but with greater flexibility. At first, the program was run with the conventional settings, i.e. without considering fertility or mortality change and with only the life expectancy at birth (i.e. *parc[4]*) as a free parameter. The values of *par[1]* and *par[2]* are derived from the earlier Relational Gompertz application. These are the settings as shown in the R script above. The results (estimated life expectancy, deviance and fitted proportion of children dead by mother's age group) are the following:



```
Life Table Model 1; e0 = 73.15; Deviance = 11463;
                    fitpd = 0.0282 0.0303 0.0319 0.0331 0.0344 0.0361 0.0384

Life Table Model 2; e0 = 72.95; Deviance = 5638;
                    fitpd = 0.0252 0.0286 0.0314 0.0337 0.0362 0.0394 0.044

Life Table Model 3; e0 = 78.96; Deviance = 13321;
                    fitpd = 0.0278 0.0306 0.0323 0.0333 0.0342 0.0352 0.0366

Life Table Model 4; e0 = 71.43; Deviance = 7669;
                    fitpd = 0.0264 0.0294 0.0317 0.0335 0.0355 0.0381 0.0419

Life Table Model 5; e0 = 78.33; Deviance = 14070;
                    fitpd = 0.0285 0.0308 0.0322 0.0331 0.0339 0.0349 0.0362

Life Table Model 6; e0 = 72.20; Deviance = 10133;
                    fitpd = 0.0273 0.0302 0.0322 0.0336 0.035 0.0369 0.0397

Life Table Model 7; e0 = 75.83; Deviance = 10760;
                    fitpd = 0.0268 0.0301 0.0323 0.0338 0.0352 0.0367 0.0387

Life Table Model 8; e0 = 77.57; Deviance = 10403;
                    fitpd = 0.0262 0.0298 0.0323 0.034 0.0354 0.0368 0.0387

Life Table Model 9; e0 = 78.42; Deviance = 12536;
                    fitpd = 0.027 0.0303 0.0324 0.0338 0.0348 0.0358 0.0369
```

The model that best represents the data is number 2 (Princeton North), which results in a female life expectancy at birth of 72.95 years. Its deviance, however, is 5638 and the *fitpd* series increases too slowly with age, suggesting that there may be room for improvement.

Next, *par[3]* was set at a value of 1.0248, i.e. the fertility decline implied by the Relational Gompertz application in Figure 2, and the *par[1]* and *par[2]* parameters were reset accordingly. This changes the results remarkably little. The deviance for the North model decreases marginally, to 5285, and the estimated life expectancy increases slightly, to 73.12 years.

In the next step, all three parameters *par[4]*, *par[5]* and *par[6]* were released. The age pattern of excess mortality *excess*, which was not relevant so far, now acquires some importance. For lack of a better criterion, it was fixed at the admittedly somewhat arbitrary values shown in the R script. The release of all three parameters drastically reduces the deviance, to the 106-118 range. The fact that there is no clear favorite makes it difficult to decide which model to choose, especially as the implied life expectancies vary considerably, from 75.73 years in the Princeton West model to 84.45 years in the Priceton South model. This specification also leads to a rather large implied mortality decline, with *parc[5]* values for all models in excess of 1.038. This is higher than the average annual reduction factor of mortality (death rates between ages 0 and 35) between 1988 and 2018 estimated by the UN Population Division (2022), which is 1.0335.

Finally, therefore, *parc[5]* was fixed at the more realistic value of 1.0335. In this case, the Princeton North model provides the best fit, with a deviance of 180, resulting in an estimated life expectancy of 77.86 years. The second best is the Princeton West model, with a deviance of 317 and an estimated life expectancy of 74.45 years. The large variation in estimated life expectancies is due, of course, to the fact that these are all extrapolations from infant and child mortality according to different life table models, without any consideration of adult mortality. The UN Population Division, which did make this more comprehensive assessment, estimated the 2018 female life expectancy for Guatemala in 2018 as the 75.6 years, i.e. in between the vales extrapolated from a Princeton North and Princeton West model of infant and child mortality.

Given the fact that the observed proportions of children (girls) that died refers to an average over roughly a 25 year time period, it should come as no surprise that the higher the decline of mortality is assumed to be, the lower the estimated current mortality and the higher the estimated life expectancy will be (Kraly and Norris, 1978). This makes it all the more important not to assume that mortality levels have been constant, but to use realistic estimates of *par[5]*, as the resulting estimates are sensitive to this assumption.

Final considerations

As was noted in the Introduction, the illustrative programs provided here could be made more user-friendly and more useful for actual applications. Rather than focusing narrowly on the essence of the Sprague method, with only five age intervals, the program could be made easier to use by allowing for additional age intervals. Outputs could be made more comprehensive and formatted more intuitively. The pattern of fertility and mortality change, as well as the impact of excess mortality in the past or for the lowest ages of mothers could allow for alternative options, other than those chosen in the last two examples. The somewhat confusing treatment of the age variables, due to the fact that R does not allow vectors to start at indices other than 1, could be resolved more transparently. Similarly, there may be a need for more detailed instructions on how to use the extensions proposed in this article and more examples could be added. In the future these might be converted into an actual software package. However, because the purpose of the examples is illustrative, such practical improvements or further explanations were not pursued here.

Ultimately, this article has illustrated how the more efficient use of the resources available in packages such as R can make several demographic methods more direct, intuitive, transparent, flexible, extendable and precise. In this age of sophisticated computational software, it no longer makes sense to program demographic methods in ways that were useful in the age of the sliderule and the pocket calculator, but that have lost their relevance in an age in which many of the once formidable algorithmic obstacles no longer exist.

References

Alexander, M., Yeung, J. & Riffe, T. (2022). Rogers Castro migration models with rcbayes. https://cran.r-project.org/web/packages/rcbayes/vignettes/intro_to_rcbayes.html

Alkema, L., Zhang, S., Chou, D., Gemmill, A, Moller, A., Ma Fat, D., Say, L., Mathers, C. & Hogan, D. (2017). A Bayesian approach to the global estimation of maternal mortality. *Annals of Applied Statistics*, *11*(3), 1245-1274. https://doi.org/10.1214/16-aoas1014

Baker, J., Alcantara, A. & Ruan, X. (2011). A stochastic version of the Brass PF ratio adjustment of age-specific fertility schedules. *PLOS One*. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0023222; https://doi.org/10.1371/journal.pone.0023222

Brass, W. (1971). On the scale of mortality. In: *Biological Aspects of Demography*. Symposia of the Society for the Study of Human Biology, Vol. 2. London: Taylor and Francis, Ltd.

Coale, A. J. (1957). A new method for calculating Lotka's r – the intrinsic rate of growth in a stable population. *Population Studies*, 1, 92-94. https://doi.org/10.2307/2172513

Dublin, L. I. & Lotka A. J. (1925). On the true rate of natural increase. *Journal of the American Statistical Association*, *20*(150), 305-339. https://doi.org/10.2307/2965517

Grupo de Foz (2021). *Métodos demográficos: uma visão desde os países de língua portuguesa*. São Paulo, Editora Blucher. https://

openaccess.blucher.com.br/article-list/9786555500837-504/list#undefined; https://doi.org/10.5151/9786555500837-18

Kraly, E. P. & Norris, D. A. (1978). An evaluation of Brass mortality estimates under conditions of declining mortality. *Demography*, *15*(4), 549-557. https://doi.org/10.2307/2061206

Mazza, A. & A. Punzo (2011). Discrete beta kernel graduation of age-specific demographic indicators. In: *New Perspectives in Modeling and Data Statistical Analysis*. New York, Springer, p. 127-134. https://doi.org/10.1007/978-3-642-11363-5_15

Monteiro da Silva, J. (s.d.). fertGompPF: Gompertz PF fertility estimation. In: *fertestr. Demographic Tools for the Assessment of Fertility and Parity Data*. https://rdrr.io/github/ josehcms/fertestr/

Moultrie, T. A., Dorrington, R. E., Hill, A. G., Hill, K. H., Timæus, I. M. & Zaba, B. (Eds.). (2013). *Tools for Demographic Estimation*. Paris: International Union for the Scientific Study of Population, Chapter 7. https://demographicestimation.iussp.org/

Raftery, A. E., Alkema, L. & Gerland, P. (2014). Bayesian population projections for the United Nations. *Statistical Science*, *29*(1), 58-68. https://doi.org/10.1214/13-sts419

Riffe, T., Aburto, J. M., Alexander, M., Fennell, S., Kashnitsky, I., Pascariu, M. & Gerland, P. (2019). *DemoTools: An R Package of Tools for Aggregate Demographic Analysis* https://github.com/timriffe/DemoTools/

Rizzi, S., Gampe, J. & Eilers, P. H. C. (2015). Efficient estimation of smooth distributions from coarsely grouped data. *American Journal of Epidemiology*, *182*(2), 138-147. https://doi.org/10.1093/aje/kwv020

Siegel, J. S. & Swanson, D. A. (2004). *The Methods and Materials of Demography*. San Diego CA: Elsevier Academic Press.

UN Population Division. (1983). *Manual X: Indirect Techniques for Demographic Estimation*. New York: United Nations, DESA. https://unstats.un.org/unsd/demographic/standmeth/handbooks/Manual_X-en.pdf

UN Population Division. (2022). *World Population Prospects: The 2022 Revision*. New York: United Nations, DESA. https://doi.org/10.18356/9789210014380

Wilmoth, J., Zureick, S., Canudas-Romo, V., Inoue, M. & Sawyer, C. (2012). A flexible two-dimensional mortality model for use in indirect estimation. *Population Studies*, *66*(1), 1-28. https://doi.org/10.1080/00324728.2011.611411

Zaba, B. (1981). *Use of the Relational Gompertz Model in Analysing Fertility Data Collected in Retrospective Surveys*. London: London School of Hygiene and Tropical Medicine.