



The R-scripts and numerical results in the prior chapters assumed that investors in the modern tontine live and die by the famous Benjamin Gompertz (1825) model of mortality. And, although that *natural law* is about to celebrate its bicentennial and has withstood the test of time and longevity, there is a growing body of (controversial) evidence the law is violated at advanced ages. More importantly, practising real-world actuaries are accustomed to using a vector of discrete  $q_x$  mortality tables and many are sceptical the force of mortality can be treated as smooth and differentiable. (Even more distressing, some have never heard of Gompertz.) The good news is that almost any discrete mortality assumption can be easily and quickly *plugged into* the simulation R-scripts presented in the earlier chapters—and that will be explained. Another important issue addressed in this chapter is how and when changes to mortality models themselves might affect the evolution and stability of tontine dividends. Essentially, the plan in this chapter is to kill people very differently, otherwise known as model risk. To start though, we won't kill anyone at all and I'll begin by introducing the non-tontine version of a *decumulation* fund.

## 7.1 Assumptions Versus Realizations

As noted above, the focus and purpose of this chapter is to measure and monitor the impact of different mortality *assumptions* and *realizations* on the long-term behaviour of the modern tontine fund and its dividends. To be more specific, if the sponsor believes investors will be healthier, dying later and living longer, the initial payout rate must be reduced in order to maintain tontine dividend stability. In our language, they should use a higher  $m$  value. Indeed, if an incorrect initial assumption is made about mortality and the initial payout rate is set too high, the realized tontine dividends will be forcefully adjusted downwards over time via the

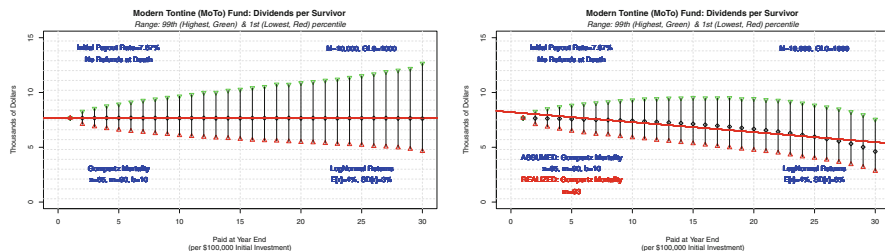


Fig. 7.1 Mortality models: assumed versus realized (98% C.I.)

thermostat mechanism. This is what I mean by the term initial *assumptions* versus actual *realizations*. The distinct between ex ante and ex post is critical.

Indeed, it’s hard to over-emphasize the importance of “getting” the mortality and longevity parameters “just right”, and the following graphical example should help illustrate what happens when wrong (or bad) assumptions are made. The left-hand panel in Fig. 7.1 displays the (by now standard) evolution of tontine dividends *assuming* that—and *realizing* a future in which—mortality evolves according to a Gompertz model with modal value  $m = 90$  and dispersion value  $b = 10$  years. Under a valuation (and expected return) of 4%, the tontine dividends propagate in a relatively symmetric manner over time with an expanding band of uncertainty caused by the diminishing number of survivors. The relatively (high) initial payout rate of  $\kappa_1 = 7.67\%$  assumes no cash refund at death or any other covenants. This is the original  $v1.0$  introduced and explored many chapters ago and should be routine.

In contrast to the usual picture, the right-hand panel of Fig. 7.1 makes one small and minor change to the canonical R-script. Although the initial payout rate and the  $\kappa$  curve is computed based on the parameter:  $m = 90$ , the simulation of deaths via the R-script line: `rbinom(1, GLIVE[i, j - 1], 1 - TPXG(x + j - 1, 1, 93, b))` assumes that mortality rates are (lower) and driven by a modal value of  $m = 93$ . In other words, people die at a slower rate, there are more survivors than expected and tontine dividends must gradually be reduced relative to what was projected and promised.

Notice how the cone and envelope of uncertainty begins to trend downwards over time as more people (than expected) survive. While the total amount (numerator) paid out to survivors as a fraction of the fund value remains the same—determined by the vector of  $\kappa$  values—the number of survivors (denominator) is larger than anticipated and the tontine dividends shrink. Either way this is a simple example and illustration of what happens if life expectancy (or more precisely, the modal value of life) ends up being a mere 3 years greater than anticipated and mortality rates are systematically lower across the board. The death rate at age  $y$  was assumed to be:  $(1 - TPXG(y, 1, 90, b))$ , when it should have been:  $(1 - TPXG(y, 1, 93, b))$ . Mortality modelling is quite important to the life of a *modern tontine* fund, and you already heard something similar in Chap. 4.

## 7.2 No Mortality: Natural Decumulation

And yet, before we continue along the journey of investigating the sensitivity of the modern tontine to other mortality assumptions and forms, I would like to proceed with a hypothetical in which there is no mortality assumed or realized at all. In other words, I would like to examine the operations of the *anti-tontine fund* in which the initial capital and all market returns are returned to all investors (who never die) over the entire horizon of the fund. I'll call this the *Natural Decumulation (NaDe) Fund* to contrast it with the *Modern Tontine (MoTo)*. Operationally this will allow beneficiaries or their estate to inherit shares of the *natural decumulation fund* and continue the same cash-flow stream. The point of this odd diversion is to investigate how a twin fund without any longevity risk sharing or pooling might behave, to then help contrast with its super-charged sibling, shedding a new unique light on mortality assumptions. Alas, it's not inconceivable that such a fund might be offered to help highlight the greater longevity and mortality credits embedded within its sister.

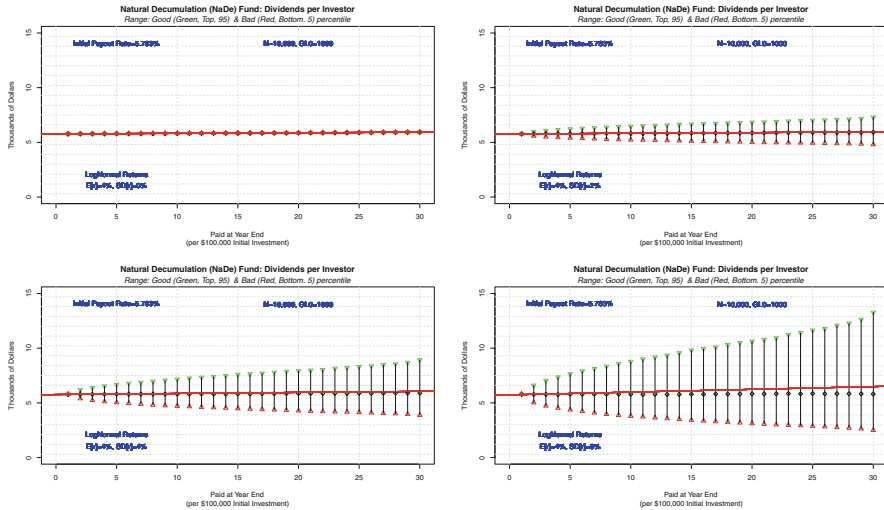
A quick-and-easy way to simulate a *natural decumulation (NaDe)* fund would be to use our existing R-script and replace the modal age at which people die, that is the Gompertz  $m$  parameter with a ridiculously high number (like a million years). That would certainly ensure that the *force of mortality* in the next few centuries is forced down to zero and nobody is *assumed* to die or *actually* dies. Although that fix would certainly do the trick, it would create a very large and unnecessary number of matrices, tables and variables. It would also slow down the code. After all, what's the point of having a GDEAD full of zeros, or GLIVE with the same people as there were on day one. Instead, I will take this opportunity to create a slimmed down R-script, one that allows us to compare and contrast *modern tontines* against *natural decumulation funds*. Here is the script.

```
set.seed(1693)
# Natural Decumulation Fund (non-tontine.)
RGOA<-function(g,v,N){(1-((1+g)^N)*(1+v)^(-N))/((v-g)/(1+g))}
TH<-30; N<-10000; EXR<-0.04; SDR<-0.03;
r<-0.04; f0<-100; GL0<-1000
kappa<-c()
for (i in 1:TH){kappa[i]<-1/RGOA(0,r,(TH-i+1))}
PORET<-matrix(nrow=N,ncol=TH)
DECFV<-matrix(nrow=N,ncol=TH)
DECDV<-matrix(nrow=N,ncol=TH)
for (i in 1:N){PORET[i,]<-exp(rnorm(TH,EXR,SDR))-1}
for (i in 1:N){
  DECDV[i,1]<-f0*kappa[1]
  DECFV[i,1]<-f0*GL0*(1+PORET[i,1])-DECDV[i,1]*GL0
  for (j in 2:TH){
    DECDV[i,j]<-DECFV[i,j-1]*kappa[j]/GL0
    DECFV[i,j]<-DECFV[i,j-1]*(1+PORET[i,j])-DECDV[i,j]*GL0}}
}
```

I'll now go through the R-script line-by-line and explain what each piece is doing. Many of the ingredients should be familiar by now. The very first line defines a new function `RGOA`, which is an abbreviation for regular growth ordinary annuity. It represents the present value of \$1, growing at the rate of  $g$  per period, received over  $N$  periods, where the discount or valuation rate is  $v$ . So, for example, `RGOA(0, 0.04, 30)` leads to a present value factor of: \$17.292, and the more important inverse of this number is 5.783%. What this effectively means is that an initial investment of \$100,000 is economically equivalent to a constant payment of \$5,783 per year for 30 years. The present value of \$5,783 annually for the next 30 years is equal to \$100,000. If this reminds you of the  $\kappa$  function and its inverse the annuity factor, that is exactly my point. The payout rate of the non-tontine will be based on this function so when valuation rates are 4%, the initial payout rate from the *natural decumulation* fund will be 5.783%, versus the above-noted 7.67% for the *modern tontine*. The 189 basis point difference between these two numbers and payouts is due to: *mortality credits*. In some sense, the gap or spread—regardless of the parameters—highlights what tontine participants are getting in return for sacrificing their principal.

With that one line of R-script out of the way, we can move a bit faster down the remaining ones. The parameters are defined (with no mortality!) and then the  $\kappa$  vector is defined as the reciprocal of the `RGOA` function, instead of the Gompertz-based `TLIA` or `RTLIA` functions. The code names the same (old) `PORET` matrix to capture and simulate the future investment returns but defines new `DECFV` and `DECDV` to keep track of the natural decumulation fund value and the dividends, which should probably be called liquidation payouts or blended return of principal and investment gains. The word dividend is pushing the nomenclature a bit far. Nevertheless, the process of populating those two matrices is quite similar to its twin tontine sibling, except that the distributable quantity `DECFV[] * kappa[]` is divided by the same original number of investors `GL0`, not the shrinking `GLIVE[]` matrix.

For example, let's use a seed of 1693 and compute the year-20 mean and standard deviation of the modern tontine dividends using `v1.0` of the R-script and compare those with the mean and standard deviation of the non-tontine, a.k.a. *natural decumulation* fund. Use the familiar  $m = 90$ ,  $b = 10$ , as well as  $r = v = 4\%$  with a standard deviation  $\sigma = 3\%$ , and make certain you get these exact numbers, in the process confirming your (new and old) code is working correctly. Recall that in the later part of this book we have migrated towards a higher value of  $\sigma$ , to be able to afford (and justify) a higher portfolio expected return and discount rate. The mean tontine dividend in year 20 is \$7,750 per initial \$100,000 investment, and the standard deviation (scaled by the mean) is: `sd(TONDV[, 20]) / mean(TONDV[, 20])`, which is: 14.76%. Alas, for the non-tontine fund, the mean dividend in year 20 is: `mean(DECDV[, 20]) = 5.9115` thousand dollars, and the standard deviation is: `sd(DECDV[, 20]) / mean(DECDV[, 20])`, or 14.12%, a much lower mean and a slightly lower standard deviation.



**Fig. 7.2** Natural decumulation fund dividends: 0%, 2%, 4% and 8% vol

Finally, Fig. 7.2 displays the non-tontine dividends over time assuming the usual stochastic (LogNormal) investment returns, but with no other source of uncertainty; certainly no deaths or lapsation. Rather, every single year the entire value of the fund is effectively *divided* by the relevant annuity factor and distributed to all investors. Although I have displayed the range of payouts under four different volatility levels, one thing that does seem rather clear is that *even* without any mortality uncertainty—or any relevance of death—there is an increased variability at advanced ages and periods. You can’t blame a small surviving pool for that. That band is due to investment returns and markets. To wrap this all up, at the very end of the script you could also add the usual regression to test stability of the *average* payouts over time.

```
mtd<-c(); t<-1:TH
for (j in 1:TH){mtd[j]<-median(DECLDV[,j])}
fit<-lm(mtd~t); summary(fit)
```

### 7.3 Isolating Mortality Credits

The point of a *natural decumulation* fund isn’t simply to act as a foil or baseline for the *modern tontine*, but in fact might be a viable product that is offered for sale at the same time. Retirees—or perhaps better described as decumulators—would have a choice of investing in either the *modern tontine*, which I’ll abbreviate with MoTo or the *natural decumulation* fund NaDe. The point of offering these two funds together and at the same time would be to highlight the value or benefits from pooling longevity risk. Of course, the initial yield or payout from MoTo would

be higher than NaDe, and the exact difference between the two would (obviously) depend on mortality assumptions, which is something I'll return to in just a bit.

---

## 7.4 Fitting Gompertz at the Table

The time has come for me to address the elephant in the room (or this book), which is my continuous use of the historic 1825 Benjamin Gompertz law of mortality, when most actuaries in the twenty-first century use discrete mortality tables—and many of them—to value, price and reserve against life annuity liabilities. Indeed, industry actuaries are only vaguely familiar with or aware of Gompertz, last seen during their actuarial exams in school. Rather, the common practice in the insurance industry is to fix a proper *mortality basis* by selecting a (i.) suitable mortality table with possible (ii.) improvement factors, and then combining those with a valuation rate assumption to compute actuarial present values. So, I now discuss how to reconcile those two distinct approaches within the context of *modern tontine* simulations.

For those who might be new to matters of life and death, a mortality table is a set of numbers ranging from zero to one, which represent the fraction of individuals—alive on their birthday—who are *expected* to die prior to their next birthday. So, for example, one element in a mortality table might be  $q_{65} = 0.01$ , which should be interpreted to mean that one percent of a sufficiently large group of 65-year-olds is expected to die before their 66th birthday. A full mortality table contains many such mortality rates, ranging from a very young age to an advanced age. These  $q_x$  numbers increase and eventually hit one. Eventually everyone dies.

In what follows I will work with a particular mortality table, the CPM2014 table, although everything I do over the next few pages can be applied to anyone of the hundreds (perhaps even thousands) that are used by actuaries on a daily basis. I have selected the CPM2014, which again can be obtained from my website [www.MosheMilevsky.com](http://www.MosheMilevsky.com) if you can't find it anywhere else, because it is often used to model the mortality of pensioners and future retirees in Canada. It's suitable for the audience who might be interested in *modern tontines*.

Once you have located the data (online) and saved (in your own unique folder), import the mortality table by running the following R-script or using the standard graphic interface in R-studio. Please `view` the file once it has been imported, a part of which I have displayed in Fig. 7.3.

```
library(readr)
CPM2014 <- read_csv("~/CPM2014.csv")
View(CPM2014)
```

After the row index counter in **R**, you will see four columns. The first is an age that ranges from 18 to 115. The second column is the one-year mortality rate for males, the third column is for females and the fourth column is a unisex blended average of the two genders. The unisex mortality rates are the *equally* weighted

**Fig. 7.3** A corner section of the CPM2014 (mortality) table

	Age	qx_m	qx_f	qx_u
47	64	0.00790	0.00511	0.00651
48	65	0.00844	0.00562	0.00703
49	66	0.00907	0.00617	0.00762
50	67	0.00981	0.00675	0.00828
51	68	0.01066	0.00739	0.00903
52	69	0.01166	0.00809	0.00988
53	70	0.01282	0.00886	0.01084
54	71	0.01417	0.00973	0.01195
55	72	0.01571	0.01072	0.01322
56	73	0.01749	0.01185	0.01467
57	74	0.01952	0.01316	0.01634
58	75	0.02183	0.01469	0.01826

average of the male and female  $q_x$  rates, which can be confirmed by running the following script and should result in a large collection of zeros:

```
round((0.5)*CPM2014$qx_m+(0.5)*CPM2014$qx_f-CPM2014$qx_u,4)
> 0...
```

In the future we might want to generate our own *biased* unisex averages that differ from the fourth column, perhaps based on realized purchase experience. That average might tilt the genders in one way or the other depending on the distribution of investors in the tontine scheme. For now though, I will work with what’s given—and focus on the unisex vector—and simply ask you to keep in mind that it doesn’t necessarily have to imply a 50/50 split. Next, I will reduce the length of the name of the variable by defining a new  $qx$  to equal the longer named CPM2014 and focus on the ages from 65 to 94, which are the 48th to 77th row. Here is the script:

```
qx<-CPM2014$qx_u[48:77]
length(qx)
> 30
min(qx)
> 0.00703
max(qx)
> 0.19541
```

Notice how the lowest (dying between age 65 and 66) mortality rate is 0.7% and the highest (dying between age 94 and 95) is 19.54%, or nearly one in 5. Remember, that  $q_x$  value is an average, assuming a large group of individuals. It’s not a guarantee, or promise. This is precisely why I have randomized the number of deaths in the GDEAD matrix, using the Binomial distribution.

Next it’s time to value temporary life annuities and compute their inverse the *initial payout yield* using the above CPM2014 mortality table. The point of this

exercise is to examine how that number compares and contrasts with an *initial payout yield* under the Gompertz model, which was our `TLIA(.)` function in Chap. 3. Without repeating the theory again, this involves computing the conditional survival probability (from age 65) to the end of the year at which \$ payments take place and then discounting those \$ payments by the valuation rate. The survival probability is the product of the one-minus mortality rate  $q_x$ , and the discounting can be achieved by creating a vector of present values. I multiply those two vectors together, sum up the discounted conditional cash-flows and compute the inverse to obtain the all-important *initial payout yield*. Here is the (short) script under a 4% effective annual valuation rate.

```
r<-0.04
# Probability of Survival to each age from the present.
ps<-cumprod(1-qx)
# Discount Rate for each year to the present.
dr<-cumprod(rep(1/(1+r),length(qx)))
# Reciprocal of Actuarial Present Value of annual $1 payments.
1/sum(dr*ps)
> 0.0741556
```

Under a 4% effective annual valuation rate, the *initial payout yield* for a 30-year temporary annuity using discrete annual and unisex mortality rates within the CPM2014 mortality table is 7.4%. This is 340 basis points above the valuation rate of 4%. That assumes annual end-of-year payments, which begin on the annuitant's 66th birthday and end on his/her 95th birthday, assuming he/she is alive on that date. For those who might be puzzled by the new commands in the short script, the `rep(.)` creates a vector that repeats  $1/(1+r)$  a total of `length(qx)` times. The `cumprod(.)` creates a new vector of the same length, in which the individual elements are cumulatively multiplied and added together. Think of that as the yield curve or term structure of discount rates. Finally, `sum(.)` adds them all together. This process can be repeated for any effective annual interest rate, and under  $r = 2\%$  the *initial payout yield* is 6%, all which you should confirm for yourself. In fact, you can also compute *initial payout yields* for other starting ages (e.g. 60 or 70) by setting the  $q_x$  vector to equal that subset of the CPM2014 table.

Now, let's compare those 7.4% and 6.0% *initial payout yields*—using the CPM2014 unisex mortality table—against the Gompertz law of mortality. For convenience, I will recreate the original script explicitly in terms of the Gompertz survival curve (from Chap. 3) with parameters  $(m, b)$ .

```
TLIA<-function(x,y,r,m,b){
  APV<-function(t){exp(-r*t)*exp(exp((x-m)/b)*(1-exp(t/b))) }
  sum(APV(1:(y-x))) }
```

It's worth looking at that **R**-script carefully once again and noting the symmetry between the APV function, which is a product of the discount factor and the survival probability, and the `sum(dr*ps)` segment in the earlier script using the discrete mortality table. Both are doing the same thing. I will now evaluate the `TLIA`



function using the same 4% and 2% interest rates, assuming the usual and familiar ( $m = 90, b = 10$ ) parameters. Note that the TLIA function takes as input a *continuously compounded* interest rate  $r$ , so I will be using  $\ln[1.04]$  and  $\ln[1.02]$ , which recall is the `log` command in **R**. Finally, here are the results:

```
1/TLIA(65, 95, log(1.04), 90, 10)
> 0.07610133
1/TLIA(65, 95, log(1.02), 90, 10)
> 0.06177168
```

Under the 4% valuation interest rate, the *initial payout rate* under the Gompertz model—relative to unisex CPM2014 mortality—is approximately 19 basis points higher, and 7.61% versus 7.42%. Gompertz is more generous and pays more because he is more deadly. Under the lower and more conservative expected rate of return of 2%, the *initial payout rate* is about 16 basis points higher under Gompertz. Once again, the mortality credits are more generous. Slightly more people are assumed to die under the Gompertz ( $m = 90, b = 10$ ) assumption. So, is this good news or bad news? Is this gap of 0.2% small enough?

Well, when one considers the many other sources of uncertainty including the assumed mix of genders, payment frictions and management costs, I would say the two *initial payout yields* are close enough for comfort in Gompertz. To be more precise, the Gompertz mortality assumption with parameters ( $m = 90, b = 10$ ) will slightly over-estimate mortality, assuming slightly more deaths—relative to the CPM2014 table—and earlier on, thus leading to the higher and more lucrative mortality credits. But to be very clear, and certainly from a pedagogical point of view, the results are close enough to vindicate Benjamin Gompertz within the context of annuity factor valuation. I do not think the 20 basis point difference is material.

Now, this result or effect directly depends on the Gompertz parameters themselves. For example, if I were to (erroneously) assume that (say) the modal value of life is (only)  $m = 80$  years and that the dispersion coefficient was a lower  $b = 8$  years, the *initial payout yield* using these Gompertz parameters in the `TLIA(.)` function would be quite different, and much higher than the discrete mortality table values.

```
1/TLIA(65, 95, log(1.04), 80, 8)
> 0.1057475
1/TLIA(65, 95, log(1.02), 80, 8)
> 0.0909322
```

Intuitively, a lower assumed life expectancy and a higher force of mortality, which recall is  $(1/b)e^{(x-m)/b}$  results in 250 to 300 more basis points of mortality credits. But those numbers aren't consistent or aligned with the 7.4% and 6% initial payout rates from the discrete CPM2014 mortality table.

So, to phrase the question once more, is the *continuous* Gompertz law of mortality consistent with the CPM2014 *discrete* mortality table for the purpose of

computing *initial payout yields*? Well, it depends on the actual parameters selected within the simulation model. For ( $m = 80, b = 8$ ), the answer is a definite no, as you can see. The *initial payout yields* are quite different. But, for the above-noted parameters of  $m = 90$  and  $b = 10$  the payout factors are close and the Gompertz law of mortality is a good approximation. The bad fit is due to a poor choice of *parameters* versus a poor choice of *model*. That distinction—model versus parameters—is something that’s important to remember at all times.

## 7.5 A Look Under the (Mortality) Table

Now, to help understand why the initial payout rates (a.k.a. the inverse of the annuity factors) are so close (or so far) from each other, one has to dig a little deeper and examine the underlying one-year mortality rates  $q_x$  and their cumulative survival probabilities. Recall that the vector  $\text{ps}$  contains the survival probabilities from age 65 to age 95. For completeness, I display them here to 3 digits.

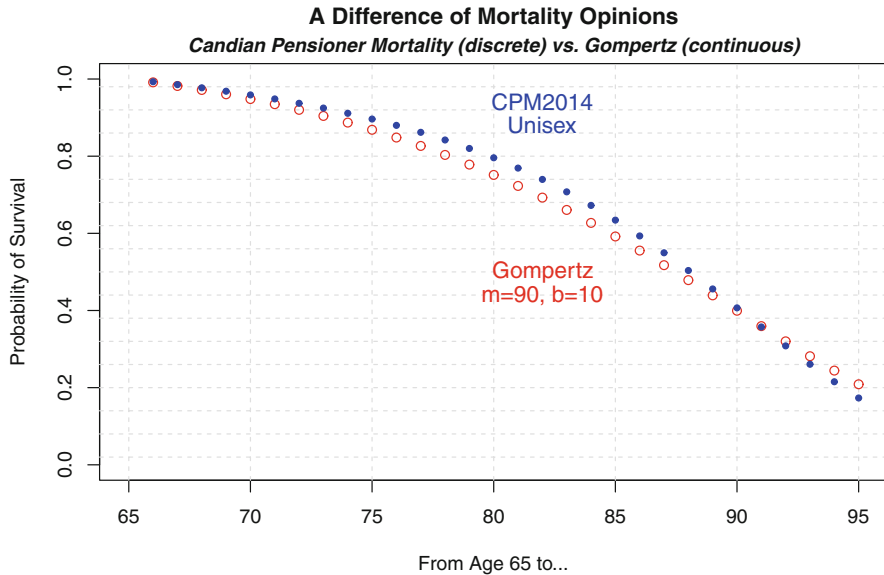
```
round(ps, 3)
0.993 0.985 0.977 0.968 0.959 0.948 0.937 0.925 0.911 0.896
0.880 0.862 0.842 0.820 0.796 0.769 0.740 0.708 0.672 0.634
0.593 0.550 0.504 0.456 0.407 0.357 0.308 0.261 0.215 0.173
```

Under the CPM2014 assumption, if we begin with 1000 (unisex) investors who are all 65 years old, we can expect to lose 7 of them in the first year, leaving 993 at the age of 66. Then, we *expect* to lose another 8 before the age of 67, leaving 985, etc. This process continues for 30 years at which point we *expect* to have 173 remaining at the age of 95. That is expected under the CPM2014 assumption. Now let’s examine the same numbers under the Gompertz ( $m = 90, b = 10$ ) mortality assumption. For convenience, I will repeat the syntax for generating those values, as well as the results.

```
TPXG<-function(x,t,m,b){exp(exp((x-m)/b)*(1-exp(t/b)))}
round(TPXG(65,1:30,90,10),3)
0.991 0.982 0.972 0.960 0.948 0.935 0.920 0.904 0.887 0.868
0.848 0.827 0.803 0.778 0.751 0.723 0.693 0.661 0.627 0.592
0.555 0.518 0.479 0.439 0.399 0.359 0.320 0.281 0.244 0.209
```

Notice how the *expected* Gompertz survivor numbers are slightly lower in the first decade or so, compared to the CPM2014 numbers. At the end of the first year, there are 991 survivors (from an initial group of 1000) compared to the 993. At the end of the second year, there are 982 compared to 985, etc. By the end of ten years, that is age 75, the Gompertz assumption is more deadly, leaving 868 survivors compared to the CPM2014, which has 896. Stated differently, the CPM2014 assumption is expected to leave 28 more people alive.

Recall the emphasis on the word *expected* since our core simulation doesn’t quite kill in a precisely Gompertzian manner but generates random deaths that are expected to match that curve. Also, and more importantly, the Gompertz assumption



**Fig. 7.4** Comparing mortality assumptions: tables versus laws

again with ( $m = 90$ ,  $b = 10$ ) lightens up a bit on the killing as time goes on, and you will notice that by age 95 there are 209 expected survivors under Gompertz (a 20.9% survival probability), versus the 173 survivors under the CPM2014 table (which is a 17.3% survival probability). All of this can be visualized with the help of Fig. 7.4, using the following script, which I have abbreviated to preserve space.

```
plot(c(65,95),c(0,1),type="n",
     xlab="From Age 65 to...", ylab="Probability of Survival")
grid(ny=18,lty=20)
for (i in 1:30){
  points(65+i,TPXG(65,i,90,10),col="red")
  points(65+i,ps[i],col="blue",pch=20)}
```

Notice how the CPM2014 table sits above the Gompertz curve for the first two decades and then falls under in the final ten years. Now, overall it's the present value of these curves that (really) matter for valuation and payouts, and that obviously depends on the valuation rates, which is another dimension to consider. As you saw earlier, the difference in the actuarial present value was no more than 20 basis points.

In some sense the above not-only vindicates using the (simple, parametric) Gompertz model but is the basis for selecting the specific ( $m = 90$ ,  $b = 10$ ) assumption within the context of modern tontines offered to (Canadian) retirees. Now, to be very clear, if these funds were to extend their horizons to 35, 40 or to the very end of the human lifecycle, the gap between these mortality assumptions would be greater. One would certainly not be entitled to set actuarial reserves or capital requirements—especially at advanced ages—using the Gompertz assumption, but

for our purposes it's sufficient. Remember if-and-when realized mortality and the number of investors dying deviates from our initial assumptions, those losses (or gains) will be accounted for in the *thermostat* design of the fund.

## 7.6 Projection Factors: Today vs. the Future

The number or date 2014 in the title of discrete mortality tables I have been using over the last few pages wasn't coincidental. In fact, it's meant to remind users that the numbers are *period* mortality rates for a specific group (pensioners) in a specific year (2014). Thus, the number  $q_{65}$  denotes the mortality rate for someone who is 65 years old in the year 2014. The number  $q_{66}$  is meant to denote the mortality rate for a 66-year-old in 2014, etc. That seems reasonable enough, but our simulations require probabilities of survival and mortality rates  $q_x$  for investors who are 65 today and will be 66 next year, 67 the year after, etc. There is a subtle difference between the two because a 66-year-old today is different from a 66-year-old next year, and the underlying  $q_x$  values might be different. Intuitively one might expect mortality rates for (say) 66-year-olds to decline ever so slightly over time, as newer cohorts (born later) are likely to be slightly healthier.

What all this means is that current (a.k.a. period) mortality rates have to be projected or reduced into the future using *improvement factors*, which is another set of numbers used by practicing actuaries together with the basic mortality tables discussed above. Now, this is certainly not the venue for an in-depth and detailed discussion of mortality improvement factors, how they are created and when they are used, but I shall say the following. It's rather easy to implement those into the existing simulation code, but they will change the results. In fact, if one assumes very large improvements—for example mortality rates declining by 5% every single year—then survival rates will be much higher, less people will die over the next 30 years and dividends will be reduced. To get a sense of how these *improvement factors* can impact mortality rates  $q_x$  and survival rates, I offer the following example.

Before you review and run the script, allow me to explain *how* I'm projecting mortality improvements on the basic CPM2014 table. I'm not using any pre-determined projection scales or factors, but I'm artificially assuming the following. Mortality rates for anyone between the age of 65 and 75 will improve (that is decline) by 3% every single year. Mortality rates for individuals between the age of 75 and 85 will improve by 2% per year, and for those between age 85 and 95 it will improve by 1% per year. Stated differently. In 5 years from now the relevant  $q_{65}$  will decline to  $q_{65}(1 - 0.03)^5$ . That is a decline from (current) 0.00988 to (future) 0.00848. So, the elements in the conditional survival probability `prod(1-qx)` must be adjusted accordingly. That is what this script is doing.

```
imfa<-c(rep(0.03,10),rep(0.02,10),rep(0.01,10))
qx<-CPM2014$qx_u[48:77]*(1-imfa)^(1:30)
ps<-cumprod(1-qx)
round(ps,3)
```

0.993	0.986	0.979	0.971	0.963	0.954	0.945	0.935	0.924	0.913
0.900	0.885	0.870	0.853	0.834	0.814	0.792	0.768	0.742	0.714
0.676	0.636	0.594	0.550	0.504	0.457	0.409	0.361	0.314	0.269

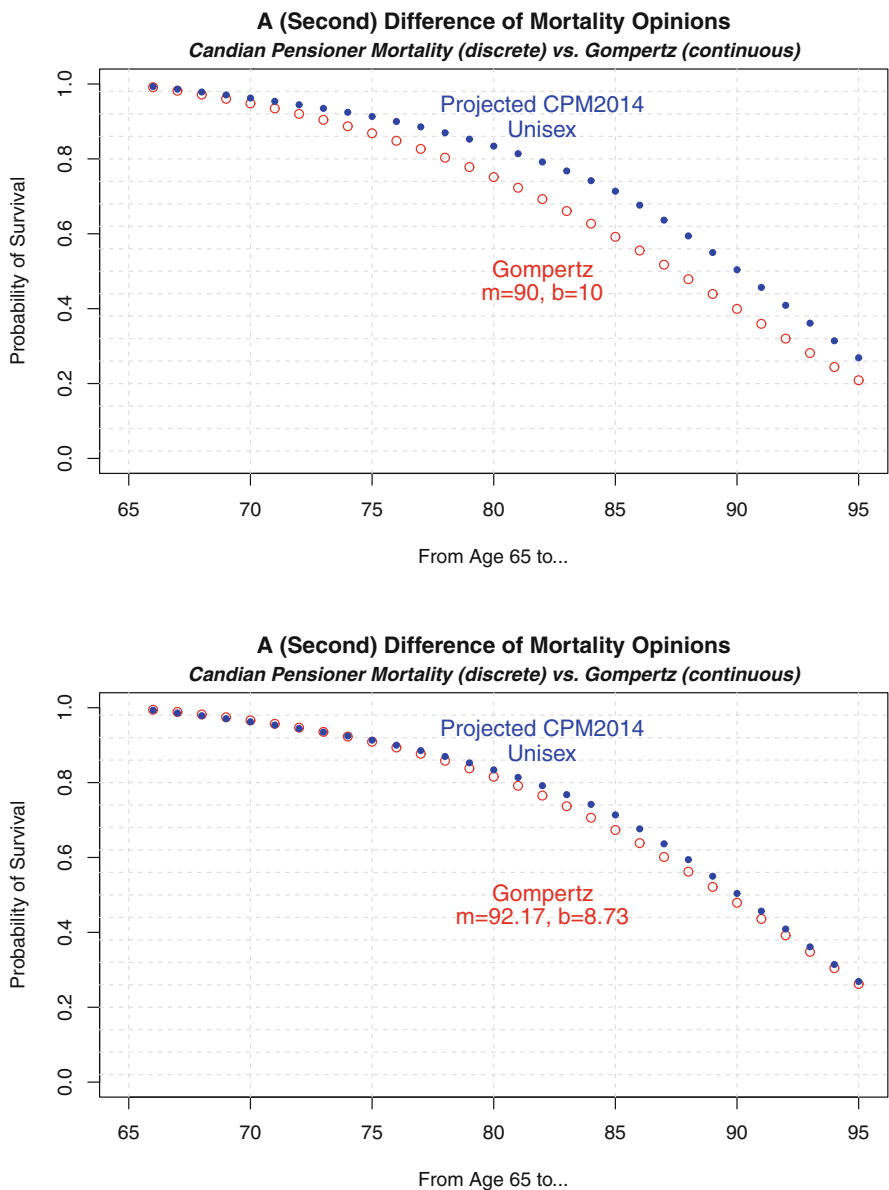
If you compare the number of survivors under the *dynamically projected* CPM2014 to the number of survivors noted earlier under the *static period* table, you will notice more survivors (starting with 1000) after year number one. For example, the above implies 986 survivors, versus 985, by the age of 67. In fact, if you look at the very last and final number, it represents 269 survivors versus the 173 without this *dynamic projection* of improvement factors. Clearly, reducing future mortality rates will have an impact on the fraction of the original 1000 that make it to the end of the 30-year period. More importantly, the Gompertz assumption with ( $m = 90, b = 10$ ) will no longer prove accurate. The *initial payout yield*, which is the inverse of the annuity factor, is now:

```
r<-0.04
# Using a Projection Scale
ps<-cumprod(1-qx)
dr<-cumprod(rep(1/(1+r),length(qx)))
round(1/sum(dr*ps),3)
> 0.071
```

This is approximately 30 basis points less than when the *static* CPM2014 was used, which is consistent with the idea that less people are dying because mortality is improving. But when compared against the  $1/TLIA(65, 95, \log(1.04), 90, 10)$  values, we are now a full half a percent (50 basis points) under the Gompertz values. Something must be done, or to be more specific and practical, the Gompertz parameters will have to be modified.

Figure 7.5 compares the *dynamically projected* values of the underlying  $q_x$  vector against the original ( $m = 90, b = 10$ ) parameters, showing the poor fit. But right under the top figure, I have plotted the Gompertz survival probability using modified parameters ( $m = 92.17, b = 8.73$ ) and now the two curves are much closer together. Intuitively if we increase the modal value of the Gompertz distribution and slightly reduce the dispersion, it has the effect of reducing mortality rates and (more) closely matching the *dynamically projected* CPM2014 table.

Now, if you happen to be wondering how (and where in the world) I came up with those two revised values of ( $m, b$ ) and how I knew they would fit better, the answer is not blind trial and error and certainly isn't divine intervention. Rather, I used a short procedure that has been fully described elsewhere—in the book *Retirement Income Recipes in R*, Chap. 8—on which I will not elaborate or repeat myself. Suffice it to say, the algorithm I used is encapsulated in the following **R**-script that uses a simple regression procedure to locate the best fitting (i.e. optimized) parameters:



**Fig. 7.5** Comparing dynamic mortality assumptions: different ( $m, b$ )

```
x<-65:94; y<-log(log(1/(1-qx)))
fit<-lm(y~x)
h<-as.numeric(fit$coefficients[1])
g<-as.numeric(fit$coefficients[2])
```

```
m<- log(g)/g-h/g; m
> 92.16698
b<- 1/g; b
> 8.728464
```

The essence of this script or its secret sauce is to linearize the mortality rate  $q_x$  via the double log calculation and then regress that number on age to obtain the Gompertz parameters based on the linearity of the term structure of mortality. If you want to understand *why* this works, the above-noted reference is the place to start. But if all you want is a quick-and-easy procedure for locating the best fitting Gompertz parameters to **any** mortality vector, then those few lines should do the trick. This trick will work whether you want to use a basic table such as the CPM2014 without any *dynamic projections* or whether you want to (first) reduce the mortality rates with unique improvement factors. The key is to fix the  $q_x$  vector and then run the script.

In fact, if you look back at Fig. 7.4 and the small gap between the CPM2014 (w/o any projection) and the Gompertz curve with parameters ( $m = 90, b = 10$ ), you might now ask yourself if we can do better and reduce the gap even further using the above-noted optimization procedure. The answer is a definite yes, and I will leave that as an end-of-chapter question. In fact, once you locate those best fitting parameters and use them to compute the *initial payout yield* using the `TLIA(.)` function, you will stumble across another extra 10 basis points, bringing you even closer to the discrete values.

---

## 7.7 Working Discretely

If I can sum up the main point of the last few pages and to conclude this chapter, it's as follows. For those users and readers who are reluctant to embrace the Gompertz assumption (and lifestyle) and would rather simulate *modern tontine* payouts using a known discrete mortality table such as the CPM2014, the solution is one line away. Instead of using and defining the survival function `TPXG`, you can define the following:

```
# Alternative to Gompertz
TPXD<-function(qx,t){prod(1-qx[1:t])}
```

Then, replace the `TPXG(.)` function, which appears in various places within the main **R**-script, with the new `TPXD(.)` script, and make sure to include the new mortality vector `qx` as one of your inputs. You can then delete and remove the `m, b` parameters within the script and forget about Gompertz altogether. In fact, the process of generating the `GDEAD[i, j]` matrix can be simplified even further, using the following substitution:

```
# Using the Gompertz Model
GDEAD[i,1]<-rbinom(1,GL0-LAPSE[i,1],1-TPXD(x,1,m,b))
```

```
# Using Discrete Mortality Table
GDEAD[i,1] <- rbinom(1, GL0-LAPSE[i,1], qx[1])
```

The reason for this is that the expression  $(1-TPXG(.))$  at the end of the `rbinom()` command is just the first year's mortality rate, which is precisely  $qx[1]$ . There is no need for the `TPXD(.)` function altogether when it comes time to kill people. The same thing would apply to the simulation of survivors in future years, in which  $(1-TPXG(x+j-1, 1, m, b))$ , which is the one-year mortality rate at age  $(x+j-1)$  is replaced with  $qx[j]$ , which does the same thing discretely. Now yes, modifying the script to run under discrete mortality will also mean removing some arguments from the `TLIA(.)` and `RTLIA` function, replacing the  $(x, m, b)$  with the vector of mortality rates  $qx$ , but that is mostly cosmetics.

In sum, think back to the triangle I presented at the very beginning of Chap. 2. While I prefer to use the Gompertz model in the *modern tontine* simulations, the upper-right corner, possibly calibrated to a mortality table using the regression procedure described earlier, those users who prefer to work directly with a discrete (and very specific) mortality table can do so directly, with just a few changes to the **R**-script. But, the main output, that is the `TONDV[i, j]` matrix, and the underlying fund values are unlikely to change very much. Of course, if you plan to create a *modern tontine* for 95-year-olds, a region of the Gompertz curve in which the fit to discrete mortality tables isn't as smooth, you might want to stick to  $q_x$  values. Then again, there are many other things to worry about if the *modern tontine* is sold to nonagenarians.

---

## 7.8 Test Yourself

- Using the non-tontine *natural decumulation* fund, please generate a table with the median as well as the (top) 99th and (bottom) 1st percentile of the fund value in 5, 10, 15 and 20 years, assuming a 4% expected return and valuation rate, and a 3% standard deviation, for a 30-year time horizon. Compare with the results for a *modern tontine* (version 1.0) and confirm that the volatility of the payouts (a.k.a. dividends) as a percent of the expected value is indeed higher for the tontine fund. Finally, force the  $m$  parameter to be (astronomically) high for the *modern tontine* script and confirm you get the same exact numerical results as the *natural decumulation* fund.
- Although the pattern of *dividends* or better described as payouts for the *natural decumulation* fund was presented within the chapter, please generate the relevant figures for the underlying fund value (using the same parameter values) and then discuss and explain their qualitative pattern.
- Investigate the impact of getting the dispersion coefficient ( $b$ ) wrong, like we did for the modal value coefficient ( $m$ ). In particular, assume some reasonable investment returns and that  $(m = 90, b = 10)$  for pricing purposes, but that realized mortality is such that  $b = 7$ . In other words, the dispersion of lifetimes



is lower, and the rate at which mortality accelerates  $1/b$  is higher than 10%. What is the qualitative impact on the pattern of modern tontine dividends over time?

4. Use the (magic) script to locate the best fitting  $(m, b)$  parameters for the basic CPM2014 mortality table at age 65, and then compute the *initial payout yield* using the Gompertz  $TLIA(.)$  function under a 4% and 2% valuation rate assumption. Compare that to the 7.4% and 6.0% yields derived and explained in the chapter.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

