The lifecontingencies Package. A Package to Perform Financial and Actuarial Mathematics Calculations in R

Giorgio Alfredo Spedicato, Ph.D

Abstract

lifecontingencies R package performs financial and actuarial mathematics calculations to model life contingencies insurance. Its functions are able to determine both the expected value and the stochastic distribution of insured benefits. Therefore they can be used both to price life insurance coverage as long as to assess portfolios' risk based capital requirements.

This paper briefly summarizes the theory regarding life contingencies, that is grounded on concepts of financial mathematics and demography. Then it shows how **lifecontingencies** package is a useful tool to perform routinary deterministic or stochastic calculations on life contingencies actuarial mathematics. Applied examples will be shown.

Keywords: life tables, financial mathematics, actuarial mathematics, life insurance.

1. Introduction

As of October 2012, **lifecontingencies** appears to be the first R package that deals with life insurance evaluation. R statistical programming environment, R Development Core Team (2012), has became the reference statistical software for academics. In the industry it is now considered a valid alternative to affirmed commercial packages for data analysis, like as as SAS,SAS Institute Inc. (2003), MATLAB, MATLAB (2010), and SPSS, Norusis (2008). With respect to the insurance industry, some actuarial packages have been already available in R, however most of these packages mainly focus non-life actuaries. In fact non - life insurance modeling uses more data analysis and applied statistical modelling than life insurance does. Functions to fit loss distributions and to perform credibility analysis are provided within the package **actuar**, Christophe Dutang, Vincent Goulet, and Mathieu Pigeon (2008). Package **actuar** represents the computational side of the classical actuarial textbook Loss Distribution, Klugman, Panjer, Willmot, and Venter (2009). The package **ChainLadder**, Gesmann and Zhang (2011), provides functions to estimate unpaid loss reserves for P&C insurances. Generalized Linear Models (GLMs), widely used in non - life insurance pricing, can be fit by functions bundled in the base R distribution. More advanced predictive models used by actuaries, that are Generalized Additive Models for Location, Shape and Scale (GAMLSS) and Tweedie Regression, can be fit using specifically developed packages as **gamlss**, Rigby and Stasinopoulos (2005), and **cplm**, Zhang (2011), packages.
Life insurance evaluation models demographic and financial data, mainly. A Finance dedicated view exists on CRAN site that lists packages specifically tailored to financial analysis.

But, few packages that handle demographic data have been published yet. For examples, relevant packages that perform demographic analysis are **demography**, Rob J Hyndman, Heather Booth, Leonie Tickle, and John Maindonald (2011), and **LifeTables**, Riffe (2011). Packages **YieldCurve**, Guirreri (2010), and **termstrc**, Ferstl and Hayden (2010), can be used to perform interest rate analysis. However, no package yet exists that performs life contingencies calculations, as of May 2012.

Numerous commercial software specifically tailored to actuarial analysis are available in commerce, on the other hand. "Moses" and "Prophet" are currently the leading actuarial software for life insurance modelling. **lifecontingencies** package aims to represent the R computational side of the concepts exposed in the classical Society of Actuaries actuarial mathematics textbook, Bowers and of Actuaries (1986). Since life contingencies theory grounds on demography and classical financial mathematics, we have made an extensive use of Chris Ruckman and Joe Francis (2006) and Broverman (2008) textbooks as references.

The paper has been structured as follows: Section 2 outlines the statistical and financial mathematics theory regarding life contingencies, Section 3 overviews the structure of the **lifecontingencies** package, Section 4 gives a wide choice of applied **lifecontingencies** examples, finally Section 5 discusses package actual and prospective development and known limitations.

2. Life contingencies statistical and financial foundations

Life insurance analysis involves the calculation of statistics regarding occurrences and amounts of future cash flows. I.e., the insurance pure premium (also known as benefit premium) is the expected value of the distribution of the insurance benefits future cash flows. Cash flows probability is based on the occurrence of the policyholder's life events (life contingencies). Therefore, life insurance actuarial mathematics grounds itself on concepts derived from demography and the theory of interest.

A life table (also called a mortality table or actuarial table) is a table that shows how mortality affects subject of a cohort across different ages. It reports for each age x, the number of l_x individuals living at the beginning of age x. It represents a sequence of $l_0, l_1, \ldots, l_\omega$, where ω, the terminal age, is the farthest age until which a subject of the cohort can survive. Life table are typically distinguished according to gender, year of birth and nationality. Life tables are also commonly developed by line of business, assurance vs annuity for example.

Using a statistical perspective, a life table allows the probability distribution of the the future lifetime for a subject aged x, to be deduced. In particular, a life table allows to derive two key probability distributions: $\tilde{T}_x$, the future lifetime for a subject aged x and its curtate form, $\tilde{K}_x$, that is the number of future years completed before death. Therefore, many statistics can be derived from the life table. A non exhaustive list follows:

- ${}_tp_x = \frac{l_{x+t}}{l_x}$, the probability that someone living at age x will reach age $x+t$.
- ${}_tq_x$, the complementary probability of ${}_tp_x$.
- ${}_td_x$, the number of deaths between age x and $x+t$.

- ${}_tL_x = \int_0^t l_{x+y}dy$, the expected number of years lived by the cohort between ages x and $x+t$.
- ${}_tm_x = \frac{{}_td_x}{{}_tL_x}$, the central mortality rate between ages x and $x+t$.
- e_x, the curtate expectation of life for a subject aged x, $e_x = E\left[\tilde{K}_x\right] = \sum_{k=1}^{\infty} {}_kp_x$.

The Keyfitz textbook, Keyfitz and Caswell (2005), provides an exhaustive coverage about life table theory and practice. Life table are usually published by institutions that have access to large amount of reliable historical data, like government statistics or social security bureaus. It is a common practice for actuaries to start from these life tables and to adapt them to the insurer's portfolio actual experience.

Classical financial mathematics deals with monetary amount that could be available in different times. The present value of a series of cash flows, reported in Equation 2, is probably the most important concept. The present value represents the current value of a series of monetary cash flows, CF_t, that will be available in different periods of time.
The interest rate, i, represents the measure of the price of money available in future times. Parallel to the interest rate, the time value of the money can be expressed by means of discount rates, $d = \frac{i}{1+i}$. This paper will use the i symbol to express the effective compound interest, when money is invested once per period. In case money is invested more frequenty, say m times per perior, each fractional period represents the interest conversion period. During each interest conversion period, the real interest rate $\frac{i^{(m)}}{m}$ is earned, where the $i^{(m)}$ expression defines the nominal rate of interest payable m times per period.

Equation 1 combines interest and discount rates, both on effective and nominal basis, to express how an amount of \$1 growths until time t.

$$A\left(t\right) = \left(1+i\right)^t = \left(1-d\right)^{-t} = \left(1+\frac{i^m}{m}\right)^{t*m} = \left(1-\frac{d^m}{m}\right)^{-t*m} \tag{1}$$

All financial mathematics functions (such annuities, $\bar{a}_{\overline{n}|}$, or accumulated values, $s_{\overline{n}|}$) can be written as a particular case of Equation 2. See the classical Broverman (2008) textbook for further reference on the topic.

$$PV = \sum_{t \in T} CF_t (1+i_t)^{-t} \tag{2}$$

Actuaries use the probabilities inherent the life table to evaluate life contingencies insurances. Life contingencies are themselves stochastic variables, in fact. A life contingencies insurance can be represented by a series of one or more payments whose occurrence and timing, and therefore their present value, are uncertain. In fact both the time and their eventual occurrence depend by events regarding the life of the policyholder (that is the reason for which they are called life contingencies). Since the actuarial analysis focuses on the present value of such uncertain payments, life contingencies insurances future payments needs to be discounted using interest rates that may be also considered stochastic. **lifecontingencies** package provides functions to model many of such random variables, $\tilde{Z}$, and in particular their expected value, the Actuarial Present Value (APV). APV is certainly the most important statistic for $\tilde{Z}$ variables that actuaries use, since it represents the average cost of the benefits the insurer guarantees to policyholders. In a P&C context it would be also known as pure premium. The benefit premiums plus the loading for expense, profits and taxes sum up to the commercial premium policyholders pay. Life contingencies can be either continue or discrete. **lifecontingencies** package models only discrete life contingencies, that is insured amounts are supposed to be due at the end of each year or fraction of year. However most continuous time life contingencies insurance are easily derived from the discrete form under broad assumptions as the Bowers and of Actuaries (1986) textbook formulas show.

Few examples of life contingencies follow:

1. An n-year term life insurance provides payment of \$ b, if the insured dies within n years from issue. If the payment is performed at the end of year of death, we can write $\tilde{Z}$ as $\tilde{Z} = \begin{cases} v^{K+1},\ K = 0, 1, \ldots, n-1 \\ 0,\ K \geq n \end{cases}$ Its APV expression is $A^{1}_{x:\overline{n}|}$.

2. A life annuity consists in a sequence of benefits paid contingent upon survival of a given life. In particular, a temporary life annuity due pays a benefit at the beginning of each period so long as the annuitant (x) survives, for up to a total of n years, or n payments. We can write $\tilde{Z}$ as $\tilde{Z} = \begin{cases} \ddot{a}_{\overline{K+1}|},\ K < n \\ \ddot{a}_{\overline{n}|},\ K \geq n \end{cases}$. Its APV expression is $\ddot{a}_{x:\overline{n}|}$.

3. An n-year pure endowment insurance grants a benefit payable at the end of n years, if the insured survives at least n years from issue. The expression of $\tilde{Z}$ is $v^{n} * I\left(\tilde{K}_{x} \geq n\right)$. Its APV expression is ${}_{n}E_{x}$.

4. A n-year endowment insurance will pay a benefit either at the earlier of the year of death or the end of the n-th year, whichever occurs earlier. We can write $\tilde{Z}$ as $\tilde{Z} = \begin{cases} v^{K+1},\ K = 0, 1, \ldots, n-1 \\ v^{n},\ K \geq n \end{cases}$. Its APV expression is $A_{x:\overline{n}|}$.

We send interested readers to the Bowers and of Actuaries (1986) textbook for formulas regarding other life contingencies insurances as $(DA)^{1}_{x:\overline{n}|}$, the decreasing term life insurance, $(IA)^{1}_{x:\overline{n}|}$, the increasing term life insurance, and common variations on payment form arrangements like deferment and fractional payments. Similarly it is possible to define insurances and annuities depending on the survival status of two or more lives. A_{xy} and $\bar{a}_{\overline{xy}}$ represent respectively the two lives joint-live insurance and the two lives last-survivor annuity immediate APVs.

The **lifecontingencies** package provides functions that allows the actuary to evaluate the APV and to draw random samples from $\tilde{Z}$ distribution. The evaluation of the APV has traditionally followed three approaches: the use of commutation tables, the current payment technique and the expected value techniques.
Commutation tables extend life table by tabulating special functions of age and rate of interest. Ratios of commutation table functions allow the actuary to evaluate APV for standard insurances. The interested reader can found a comprehensive overview of this topic in Anderson (1999) paper. The **lifecontingencies** allows underlying commutation table to be printed out as further described. However, commutation table usage has become useless in computer

era. In fact they are not enough flexible and their usage is computationally inefficient. Therefore, commutation table approach has not been used within **lifecontingencies**.
The current payment technique calculates the APV of a life contingencies insurance, $\bar{z}$, as the scalar product of three vectors: $\bar{z} = \langle\langle\bar{c}\bullet\bar{v}\rangle\bullet\bar{p}\rangle$. The vector of all possible uncertain cash flows, $\bar{c}$, the vector of discount factors, $\bar{v}$ and the vector of cash flow probability, $\bar{p}$. Since the current payment technique is the the most efficient approach from a computationally side perspective, we have used this approach to evaluate APV. Finally, the expected value approach models $\bar{z}$ as the scalar product of two vector: $\bar{z} = \langle\bar{pk}\bullet\bar{x}\rangle$. $\bar{pk}$ is $Pr\left[\tilde{K}=k\right]$, that is the probability that the future curtate lifetime to be exactly k years, $\bar{x}$ is the amount of the cash flow due under the policy term if $\tilde{K}=k$. The latter approach has been used to define the probability distribution of the life contingency $\tilde{Z}$ when performing stochastic analyses.
An example will better clarify this concept. Consider an annuity due lasting n years. Its APV, $\ddot{a}_{x:\overline{n}|}$, using the commutation tables approach is reported in Equation 3, while Equation 4 reports the APV using the current payment technique. Finally, Equation 5 calculates the APV using the expected value approach.

$$APV = \frac{N_x - N_{x+n}}{D_x} \tag{3}$$

$$APV = \sum_{k=0}^{\min(\omega-x,n)} {}_kp_x * v^k \tag{4}$$

$$APV = \sum_{k=0}^{\omega-x} \Pr\left[\tilde{K}_x = k\right] * \ddot{a}_{\overline{\min(k,n)}|} \tag{5}$$

3. The structure of the package

Package **lifecontingencies** contains classes and methods to handle life-tables and actuarial tables conveniently.

The package is loaded within the R command line as follows:

```
R> library("lifecontingencies")
```

Two main S4 classes, Chambers (2008), have been defined within the **lifecontingencies** package: the `lifetable` class and the `actuarialtable` class. The lifetable class is defined as follows

```
R> showClass("lifetable")
```

```
Class "lifetable" [in ".GlobalEnv"]

Slots:
```

```
Name:           x         lx      name
Class:    numeric    numeric character

Known Subclasses: "actuarialtable"
```

Class `actuarialtable` inherits from `lifetable` class adding one more slot dedicated to the interest rate.

```
R> showClass("actuarialtable")

Class "actuarialtable" [in ".GlobalEnv"]

Slots:

Name:   interest          x         lx      name
Class:   numeric    numeric    numeric character

Extends: "lifetable"
```

Beyond generic `S4` classes and method there are three groups of functions, reported in Table 1, Table 2 and Table 3: demographics functions, financial mathematics functions and actuarial mathematics functions. Finally, Table 4 shows **lifecontingencies** package parameters' convention.

function	purpose
`dxt`	deaths between age x and $x+t$, ${}_td_x$.
`pxt`	survival probability between age x and $x+t$, ${}_tp_x$.
`pxyzt`	survival probability for two (or more) lives, ${}_tp_{xy}$.
`qxt`	death probability between age x and $x+t$, ${}_tq_x$.
`qxyzt`	death probability for two (or more) lives, ${}_tq_{xy}$.
`Txt`	number of person-years lived after exact age x, ${}_tT_x$.
`mxt`	central death rate, ${}_tm_x$.
`exn`	expected lifetime between age x and age $x+n$, ${}_ne_x$.
`rLife`	sample from the time until death distribution underlying a life table.
`rLifexyz`	sample from the time until death distribution underlying two or more life.
`exyz`	n-year curtate lifetime of the joint-life status.
`probs2lifetable`	life table l_x from raw one - year survival / death probabilities.

Table 1: **lifecontingencies** functions for demographic analysis.

function	purpose
`presentValue`	present value for a series of cash flows.
`annuity`	present value of a annuity - certain, $a_{\overline{n}\rceil}$.
`accumulatedValue`	future value of a series of cash flows, $s_{\overline{n}\rceil}$.
`increasingAnnuity`	present value of an increasing annuity - certain, IA_n.
`decreasingAnnuity`	present value of a decreasing annuity, $DA_{\overline{n}\rceil}$.
`nominal2Real`	conversion from nominal to real interest (discount) rate.
`real2Nominal`	`nominal2Real` inverse.
`intensity2Interest`	conversion to intensity of interest from the interest rate.
`interest2Intensity`	`intensity2Interest` inverse.
`duration`	dollar / Macaulay duration of a series of cash flows
`convexity`	convexity of a series of cash flows.

Table 2: **lifecontingencies** functions for financial mathematics.

function	purpose	APV symbol
`Axn`	one life insurance	$A^1_{x:\overline{n}\rceil}$.
`AExn`	the n-year endowment	$A_{x:\overline{n}\rceil}^{\ \ 1}$.
`Axyzn`	two lives life insurances	$\bar{A}^1_{\overline{xy}:\overline{n}\rceil}$.
`axn`	one life annuity	$\ddot{a}_x$.
`axyzn`	two lives annuities	$\ddot{a}_{xy}$.
`Exn`	pure endowment	${}_nE_x$.
`Iaxn`	increasing annuity	Ia_x.
`IAxn`	increasing life insurance	$(IA)^1_{x:\overline{n}\rceil}$.
`DAxn`	decreasing life insurance	$(DA)^1_{x:\overline{n}\rceil}$.

Table 3: **lifecontingencies** functions for actuarial mathematics.

parameter	significance
`x`	the policyholder's age.
`n`	the coverage duration or payment duration.
`actuarialtable`	the actuarial table.
`i`	interest rate, that could be varying.
`k`	the frequency of payments.

Table 4: **lifecontingencies** functions parameters naming conventions.

4. Code and examples

The example secton of this paper is structured as follows: Section 4.1 deals with classical financial mathematics, Section 4.2 deals with creating and managing life tables and actuarial tables, Section 4.3 deals with classical actuarial mathematics while Section 4.4 presents the **lifecontingencies** packages functions to perform simulation analysis.

4.1. Classical financial mathematics example

The **lifecontingencies** package provides functions to perform classical financial mathematics calculations. Examples that follows show how to handle interest and discount rates with different compounding frequency, how to perform present value, annuities and future values analysis calculations as long as loans amortization and bond pricing.

Interest rate functions

Interest rates represent the time - value of the money. However different types of rates can be found in literature. As a remark, Equation 6 displays the relationship between effective interest rate, nominal interest rate, force of interest, effective discount rate and nominal discount rate.

$$(1+i)^t = \left(1+\frac{i^{(m)}}{m}\right)^t = \exp\left(\delta t\right) = (1-d)^{-t} = \left(1-\frac{d^{(m)}}{m}\right)^{-t} \tag{6}$$

Functions `interest2Discount`, `discount2Interest`, `nominal2Real`, `real2Nominal`, `interest2Intensity`, `intensity2Interest` have been based on Equation 6 and inverse formulas implied therein. Throughout the paper interest rate is deemed effective interest rate unless otherwise stated.

As examples, functions `interest2Discount` and `discount2Interest` represent a convenient way to switch from interest to discount rates and conversely.

```
R> interest2Discount(0.03)

[1] 0.02912621

R> discount2Interest(interest2Discount(0.03))

[1] 0.03
```

Function `nominal2Real` can help to evaluate what is the effective interest rate implied in a consumer - credit loan that offers 10% nominal interest rate with quarterly compounding.

```
R> nominal2Real(i=0.10,4)

[1] 0.1038129
```

Present value and internal rate of return analysis

Performing a project appraisal means evaluating the net present value (NPV) of all projected cash flows. Code below shows an example of NPV analysis

```
R> capitals=c(-1000,200,500,700)
R> times=c(0,1,2,5)
R> presentValue(cashFlows=capitals, timeIds=times,interestRates=0.03)

[1] 269.2989
```

finally both interest rate varies and cash flows are uncertain the `presentValue` function parameter `probabilities` can be properly set as following example displays.

```
R> presentValue(cashFlows=capitals, timeIds=times,
+ F interestRates=c( 0.04, 0.02, 0.03, 0.05),
+ F probabilities=c(1,1,1,0.5))

[1] -58.38946
```

The internal rate of return (IRR) is defined as the interest rate that make the NPV zero. It is used to rank alternative projects alternatively to NPV. The following example displays how to compute IRR using **lifecontingencies** package and R functions.

```
R> getIrr<-function(p) (presentValue(cashFlows=capitals, timeIds=times,
+ F interestRates=p) - 0)^2
R> nlm(getIrr,0.1)$estimate

[1] 0.1105091
```

Annuities and future values

An annuity (certain) is a sequence of payments with specified amount that is present - value, while when it is valued at the end of the term of payment is is called future values. Code below shows examples of annuities, $a_{\overline{n}|}$, and accumulated values, $s_{\overline{n}|}$, evaluations.
The PV of an annuity immediate $100 payable at the end of next 5 years at 3% is

```
R> 100*annuity(i=0.03,n=5)

[1] 457.9707
```

while the corresponding future value is

```
R> 100*accumulatedValue(i=0.03,n=5)

[1] 530.9136
```

Annuities and future values payable k-thly (where fractional payments of $1/k$ are received for each k-th of period) can be evaluated also.

```
R> ann1<-annuity(i=0.03,n=5,k=1,type="immediate")
R> ann2<-annuity(i=0.03,n=5,k=12,type="immediate")
R> c(ann1,ann2)

[1] 4.579707 4.642342
```

`increasingAnnuity` and `decreasingAnnuity` functions handle increasing and decreasing annuities, whose APV symbols are IA_x, DA_x respectively. Assuming a ten years term and a 3% interest rate, examples of increasing and decreasing annuities follow.

```
R> incrAnn<-increasingAnnuity(i=0.03, n=10,type="due")
R> decrAnn<-decreasingAnnuity(i=0.03, n=10,type="immediate")
R> c(incrAnn, decrAnn)

[1] 46.18416 48.99324
```

The last example of this section exemplifies the calculation of the present value of a geometrically increasing annuity. If amounts increase by 3% and the interest rate is 4% and its term is 10 years, the implied present value is

```
R> annuity(i=((1+0.04)/(1+0.03)-1),n=10)

[1] 9.48612
```

Loan amortization

lifecontingencies financial mathematics functions allow to define the repayments schedule of any loan arrangement, as this section exemplifies. Let C denote the loaned capital (principal), then assuming an interest rate i, the amount due to the lender at each installment is $R = \frac{C}{a_{\overline{n|}}}$. Therefore the R_t amount repays $I_t = C_{t-1} * i$ as interest and $C_t = R_t - I_t$ as capital at each installment. The loan installment, R, is initially estimated as follows

```
R> capital=100000
R> interest=0.05
R> payments_per_year=2
R> rate_per_period=(1+interest)^(1/payments_per_year)-1
R> years=30
R> R=
+ F 1/payments_per_year*capital/annuity(i=interest,
+ F n=years,k=payments_per_year)
R> R

[1] 3212.9
```

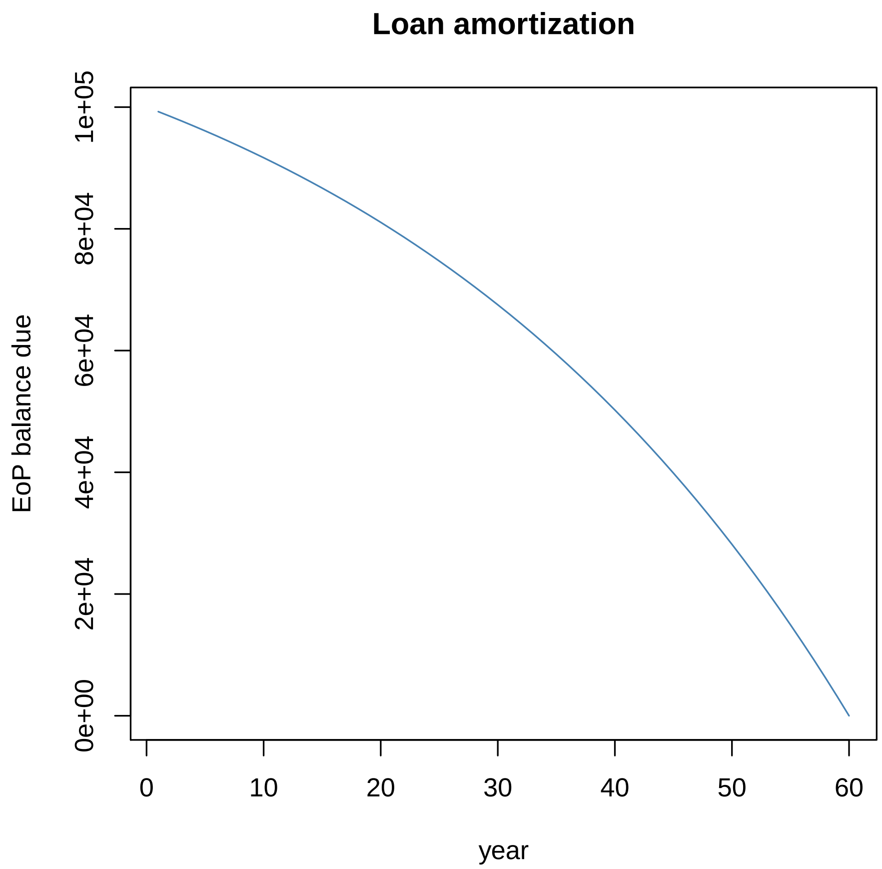

Figure 1: Loan amortization: EoP balance due.

then the balance due at end of period (EoP) is calculated as follows

```
R> balanceDue=numeric(years*payments_per_year)
R> balanceDue[1]=capital*(1+rate_per_period)-R
R> for(i in 2:length(balanceDue)) balanceDue[i]=
+ F   balanceDue[i-1]*(1+rate_per_period)-R
```

Figure 1 shows the EoP balance due for a 30 - years duration loan, assuming a 5% interest rate on a principal of \$ 100,000.

Bond pricing

Bond pricing represents another application of present value analysis. A standard bond whose face value C will be repaid at time T consists in a sequence of equal coupons c_t paid at regular intervals and a final payment of $C_T + c_T$. Equation 7 expresses the present value of a bond.

$$B_t = c_t a^{(k)}{}_{\overline{n}|} + Cv^T \tag{7}$$

Perpetuities are financial contracts that offers an indefinite sequence of payments either at the end (perpetuity-immediate) or at the beginning of the period (perpetuity-due).

Following examples show how **lifecontingencies** package elementary functions can be combined to price bond and perpetuities.

```
R> bond<-function(faceValue, couponRate, couponsPerYear, yield,maturity)
+ F {
+ F         out=NULL
+ F         numberOfCF=maturity*couponsPerYear
+ F         CFs=numeric(numberOfCF)
+ F         payments=couponRate*faceValue/couponsPerYear
+ F         cf=payments*rep(1,numberOfCF)
+ F         cf[numberOfCF]=faceValue+payments
+ F         times=seq.int(from=1/couponsPerYear, to=maturity,
+ F                 by=maturity/numberOfCF)
+ F         out=presentValue(cashFlows=cf, interestRates=yield,
+ F                    timeIds=times)
+ F         return(out)
+ F }
R> perpetuity<-function(yield, immediate=TRUE)
+ F {
+ F         out=NULL
+ F         out=1/yield
+ F         out=ifelse(immediate==TRUE,out,out*(1+yield))
+ F         return(out)
+ F }
R>
```

`bond` and `perpetuity` functions defined above can be used to price any bond, given face value, coupon rate and term, as code show displays.

```
R> bndEx1<-bond(1000,0.06,2,0.05,3)
R> bndEx2<-bond(1000,0.06,2,0.06,3)
R> ppTy1<-perpetuity(0.1)
R> c(bndEx1, bndEx2,ppTy1)

[1] 1029.250 1002.371   10.000
```

E

Duration and ALM

Duration and convexity formulas are reported in Equation 8 and Equation 9 respectively. They are used to perform portfolios asset - liability management (ALM). The interested reader could find details on Chris Ruckman and Joe Francis (2006). However, the example that follow shows how Macaulay duration (`ex1`), modified duration (`ex2`) and convexity (`ex3`) of any series of cash flows can be estimated by **lifecontingencies** package functions.

$$D = \sum_{t}^{T} \frac{t * CF_t \left(1 + \frac{i}{m}\right)^{-t*m}}{P} \tag{8}$$

$$C = \sum_{t}^{T} t * \left(t + \frac{1}{m}\right) * CF_t \left(1 + \frac{y}{m}\right)^{-m*t-2} \tag{9}$$

```
R> cashFlows=c(100,100,100,600,500,700)
R> timeVector=seq(1:6)
R> interestRate=0.03
R> ex1<-duration(cashFlows=cashFlows, timeIds=timeVector,
+ F                 i=interestRate, k = 1, macaulay = TRUE)
R> ex2<-duration(cashFlows=cashFlows, timeIds=timeVector,
+ F                 i=interestRate, k = 1, macaulay = FALSE)
R> ex3<-convexity(cashFlows=cashFlows, timeIds=timeVector,
+ F                 i=interestRate, k = 1)
R> c(ex1, ex2,ex3)

[1]  4.430218  4.563124 25.746469
```

The last example works out a small ALM problem. Suppose an insurance company has sold a guarantee term certificate (GTC) of face value $ 10,000, that will mature in 7 years at a 5% interest rate. Its final value would be:

```
R> GTCFin=10000*1.05^7
R> GTCFin

[1] 14071
```

Imagine the company can hedge its liability with two alternative investments:

1. A five year bond, with face value of 100 yearly coupon with coupon rate of 3%.
2. A perpetuity-immediate. As a remark, the formulas for the PV and Duration of the perpetuity immediate are $\frac{1}{y}$ and $\frac{1+y}{y}$ respectively when the yield is y.

In order to solve the ALM problem we need to immunize the porfolio against interest rate variation. A portfolio is immunized against parallel shift of the yield curve if both the PV and the duration of asset is set equal to the duration of liabilities. We start to figure out some parameters:

```
R> yieldT0=0.04
R> durLiab=7
R> pvLiab=GTCFin/(1+yieldT0)^7
```

```
R> pvBond=bond(100,0.03,1,yieldT0,5)
R> durBond=duration(cashFlows=c(3,3,3,3,103),timeIds=seq(1,5),i=yieldT0)
R> durPpty=(1+yieldT0)/yieldT0
R> pvPpty=perpetuity(yieldT0)
```

Then the ALM problem is set out in a three steps problem, Chris Ruckman and Joe Francis (2006):

1. setting initial the present value of cash inflows (asset) to be equal to the present value of cash outflows (liabilities).
2. setting the interest rate sensitivity (i.e., the duration) of asset to be equal to the interesr rate sensitivity of liabilities.

```
R> a=matrix(c(durBond, durPpty,1,1),nrow=2,byrow=TRUE)
R> b=as.vector(c(7,1))
R> weights=solve(a,b)
R> weights

[1] 0.8848879 0.1151121
```

Vector `weights` displays the portfolio composition in term of bonds and liabilities respectively. Therefore the number of bonds and perpetuities that can be purchased is determined by

```
R> bondNum=weights[1]*pvLiab/pvBond
R> pptyNum=weights[2]*pvLiab/pvPpty
R> bondNum

[1] 99.0279

R> pptyNum

[1] 49.23485
```

The portfolio is immunized since if interest rates suddently drops to 3% just after, the present value of assets comes to be greater than the present value of liabilities. The same occurs in case of upward shift of interest rates.

```
R> yieldT1low=0.03
R> immunizationTestLow<-(bondNum*bond(100,0.03,1,yieldT1low,5)+
+ F                          pptyNum*perpetuity(yieldT1low)>GTCFin/(1+yieldT1low)^7)
R> yieldT1high=0.05
R> immunizationTestHigh<-(bondNum*bond(100,0.03,1,yieldT1high,5)+
+ F                          pptyNum*perpetuity(yieldT1high)>GTCFin/(1+yieldT1high)^7)
R> immunizationTestLow

[1] TRUE
```

```
R> immunizationTestHigh
```

```
[1] TRUE
```

It is worth to remember that the assets allocation within the portfolio should be rebalanced since both time and changes of interest rates changes elementary securities' durations.

4.2. Life tables and actuarial tables analysis

`lifetable` and `actuarialtable` classes are designed to handle demographic and actuarial mathematics calculations. A `actuarialtable` class inherits from `lifetable` class. It has one more slot dedicated to the rate of interest. Both classes have been designed using the S4 R classes framework.
Following examples show how to initialize these classes, basic survival probabilities and life table analysis.

Creating lifetable and actuarialtable objects

Life table objects can be created by raw R commands or using existing `data.frame` objects. However, to build a `lifetable` class object three components are needed:

1. The years sequence, that is an integer sequence $0, 1, \ldots, \omega$. It shall start from zero and going to the terminal, ω, age (the age x that $p_x = 0$).
2. The l_x vector, that is the number of subjects living at the beginning of age x, that is the number of subject at risk to die between year x and $x + 1$.
3. The name of the life table.

There are three main approaches to create a `lifetable` object:

1. directly from the x and l_x vector.
2. importing x and l_x from an existing `data.frame` object.
3. from raw survival probabilities.

To create a `lifetable` object directly we can do as code below shows

```
R> x_example=seq(from=0,to=9, by=1)
R> lx_example=c(1000,950,850,700,680,600,550,400,200,50)
R> exampleLt=new("lifetable",x=x_example, lx=lx_example, name="example lifetable")
```

while `print` and `show` methods tabulate the x, l_x, ${}_tp_x$ and e_x values for a given life table.

```
R> print(exampleLt)

Life table example lifetable

  x   lx        px       ex
1 0 1000 0.9500000 4.742105
2 1  950 0.8947368 4.241176
3 2  850 0.8235294 4.042857
4 3  700 0.9714286 3.147059
5 4  680 0.8823529 2.500000
6 5  600 0.9166667 1.681818
7 6  550 0.7272727 1.125000
8 7  400 0.5000000 0.750000
9 8  200 0.2500000 0.500000
```

`head` and `tail` methods for `data.frame` S3 classes have also been implemented on `lifetable` classes

```
R> head(exampleLt)

  x   lx
1 0 1000
2 1  950
3 2  850
4 3  700
5 4  680
6 5  600
```

Nevertheless the easiest way to create a `lifetable` object is to start from a suitable existing `data.frame`. This will be probably the most practical approach for working actuaries. Some tables or mortality rates have been bundled within **lifecontingencies** package, as Table 5 displays.

data set	description
`AF92Lt`	UK AF92 life table object.
`AM92Lt`	UK AF92 life table object.
`demoChina`	China mortality rates from SOA website.
`demoIta`	Various Italian life tables including RG48 and IPS55 projected tables.
`demoJapan`	Japan mortality rates from SOA website.
`demoUsa`	US Social Security life tables.
`demoFrance`	1990 and 2002 French life tables.
`soa08`	SOA illustrative life table.
`soa08Act`	SOA illustrative actuarial table at 6%.

Table 5: **lifecontingencies** bundled life tables.

In the following example the US Social Security life tables are loaded from the existing `demoUsa` data set bundled in the **lifecontingencies** package.

```
R> data("demoUsa")
R> data("demoIta")
R> usaMale07=demoUsa[,c("age", "USSS2007M")]
R> usaMale00=demoUsa[,c("age", "USSS2000M")]
R> names(usaMale07)=c("x","lx")
R> names(usaMale00)=c("x","lx")
R> usaMale07Lt<-as(usaMale07,"lifetable")
R> usaMale07Lt@name="USA MALES 2007"
R> usaMale00Lt<-as(usaMale00,"lifetable")
R> usaMale00Lt@name="USA MALES 2000"
```

The same operation can be performed on IPS55 tables bundled in the `demoIta` data set. The purpose of following example is to stress that it is important a clean l_x series to be given in input to the coerce method. A "clean" l_x series means that neither 0 nor missing values are present anywhere and the l_x series to be decreasing.

```
R> lxIPS55M<-with(demoIta, IPS55M)
R> pos2Remove<-which(lxIPS55M %in% c(0,NA))
R> lxIPS55M<-lxIPS55M[-pos2Remove]
R> xIPS55M<-seq(0,length(lxIPS55M)-1,1)
R> lxIPS55F<-with(demoIta, IPS55F)
R> pos2Remove<-which(lxIPS55F %in% c(0,NA))
R> lxIPS55F<-lxIPS55F[-pos2Remove]
R> xIPS55F<-seq(0,length(lxIPS55F)-1,1)
R> ips55M=new("lifetable",x=xIPS55M, lx=lxIPS55M,
+ F                  name="IPS 55 Males")
R> ips55F=new("lifetable",x=xIPS55F, lx=lxIPS55F,
+ F                  name="IPS 55 Females")
```

The last way a `lifetable` object can be created is from one year survival or death probabilities combining the `probs2lifetable` function and `as.data.frame` coerce methods. Two potential applications benefit from this feature: the use of the results of a mortality projection method (e.g., the Lee - Carter method, Booth, Hyndman, Tickle, and De Jong (2006)) and the creation of "cut-down" mortality tables. The latter application is exemplified in the code line that follow where a `itaM2002reduced` life table is obtained cutting down the one - year mortality rates of Italian males aged between 20 and 60 to 20% of its original value.

```
R> data("demoIta")
R> itaM2002<-demoIta[,c("X","SIM92")]
R> names(itaM2002)=c("x","lx")
R> itaM2002Lt<-as(itaM2002,"lifetable")

removing NA and 0s

R> itaM2002Lt@name="IT 2002 Males"
R> itaM2002<-as(itaM2002Lt,"data.frame")
R> itaM2002$qx<-1-itaM2002$px
R> for(i in 20:60) itaM2002$qx[itaM2002$x==i]=0.2*itaM2002$qx[itaM2002$x==i]
R> itaM2002reduced<-probs2lifetable(probs=itaM2002[,"qx"], radix=100000,
+ F                  type="qx",name="IT 2002 Males reduced")
```

An `actuarialtable` can be easily created from a `lifetable` existing object.

```
R> exampleAct=new("actuarialtable",x=exampleLt@x, lx=exampleLt@lx,
+ F interest=0.03, name="example actuarialtable")
```

Method `getOmega` for `actuarialtable` classes returns the terminal age, ω.

```
R> getOmega(exampleAct)

[1] 9
```

Method `print` behaves differently between `lifetable` objects and `actuarialtable` objects. In fact, one year survival probability and complete expected remaining life until deaths are reported when `print` method is applied on a `lifetable` object. Classical commutation functions (D_x, N_x, C_x, M_x, R_x) are print out applying `print` method on an `actuarialtable` object.

```
R> print(exampleLt)

Life table example lifetable

  x   lx        px       ex
1 0 1000 0.9500000 4.742105
2 1  950 0.8947368 4.241176
3 2  850 0.8235294 4.042857
4 3  700 0.9714286 3.147059
5 4  680 0.8823529 2.500000
6 5  600 0.9166667 1.681818
7 6  550 0.7272727 1.125000
8 7  400 0.5000000 0.750000
9 8  200 0.2500000 0.500000

R> print(exampleAct)

Actuarial table  example actuarialtable interest rate  3 %

   x   lx         Dx         Nx        Cx       Mx        Rx
1  0 1000 1000.00000 5467.92787  48.54369 840.7400 4839.7548
2  1  950  922.33010 4467.92787  94.25959 792.1963 3999.0148
3  2  850  801.20652 3545.59778 137.27125 697.9367 3206.8185
4  3  700  640.59916 2744.39125  17.76974 560.6654 2508.8819
5  4  680  604.17119 2103.79209  69.00870 542.8957 1948.2164
6  5  600  517.56527 1499.62090  41.87421 473.8870 1405.3207
7  6  550  460.61634  982.05563 121.96373 432.0128  931.4337
8  7  400  325.23660  521.43929 157.88185 310.0491  499.4210
9  8  200  157.88185  196.20268 114.96251 152.1672  189.3719
10 9   50   38.32084   38.32084  37.20470  37.2047   37.2047
```

Finally a `plot` method can be applied to a `lifetable` or `actuarialtable` object. The underlying survival function (that is the plot of x vs l_x) is displayed in both cases. Figure 2 shows the `plot` methods applied on the Society of Actuaries (SOA) actuarial object, `soa08Act`, bundled in the **lifecontingencies** package.

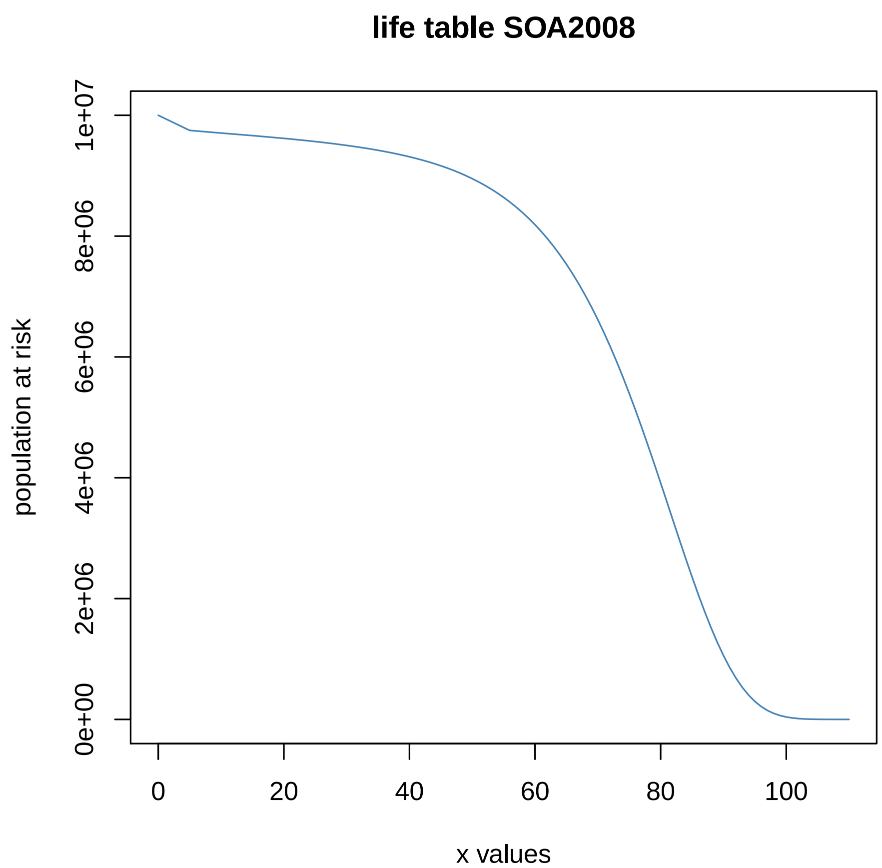

Figure 2: SOA illustrative life table underlying survival function.

Basic demographic analysis

Basic demographic estimations can be performed on valid `lifetable` or `actuariatable` objects. Code below shows how ${}_1p_{20}$, ${}_2q_{30}$ and $\mathring{e}_{50:\overline{20}|}$ respectively can be calculated on the IPS55 male population table

```
R> demoEx1<-pxt(ips55M,20,1)
R> demoEx2<-qxt(ips55M,30,2)
R> demoEx3<-exn(ips55M, 50,20)
R> c(demoEx1,demoEx2,demoEx3)

[1]  0.999595096  0.001332031 19.433217384
```

Fractional survival probabilities can also be calculated using the linear interpolation (`pxtLin`), constant force of mortality (`pxtCnst`) and hyperbolic Balducci's assumptions (`pxtHyph`), as Bowers and of Actuaries (1986) textbook details. We will show these concepts on the SOA illustrative life table, assuming insured age to be 80 years old.

```
R> data("soa08Act")
R> pxtLin=pxt(soa08Act,80,0.5,"linear")
R> pxtCnst=pxt(soa08Act,80,0.5,"constant force")
R> pxtHyph=pxt(soa08Act,80,0.5,"hyperbolic")
R> c(pxtLin,pxtCnst,pxtHyph)

[1] 0.9598496 0.9590094 0.9581701
```

Survival probabilities calculations on two lives can be performed also. As a remark, two life status are defined until multiple lives survival analysis: "joint" survival status and "last" survival status. The "joint" survival status exists until all the members are alive, while the "last" survival status exists until at least one member survives. Equation 10 defines the time until death until the joint and last survival status respectively.

$$\begin{aligned} \tilde{T}_{xy} &= \min\left(T_x, T_y\right) \\ \tilde{T}_{\bar{xy}} &= \max\left(T_x, T_y\right) \end{aligned} \tag{10}$$

Following code lines show how joint survival probabilities (`jps`), last survival probabilities (`lsp`) and expected joint lifetime (`jelt`) can be evaluated using **lifecontingencies** functions.

```
R> jsp=pxyt(ips55M,ips55F,x=65, y=63, t=2)
R> lsp=pxyt(ips55M,ips55F,x=65, y=63, t=2,status="last")
R> jelt=exyt(ips55M, ips55F, x=65,y=63, status="joint")
R> c(jsp,lsp,jelt)

[1]  0.9813187  0.9999275 19.1982972
```

4.3. Classical actuarial mathematics examples

Classical actuarial mathematics examples on life contingencies are presented. The SOA illustrative life table assuming a 6% interest rates (the same used in most Bowers and of Actuaries (1986) examples) will be used, unless otherwise stated. Similarly, the insured amount (or the annuity term payment) will be \$1, unless otherwise stated.

Life insurance examples

Following examples show the APV calculation (that is the lump sum benefit premium) for:

1. `lins1`: 10-year term life insurance for a policyholder aged 30 assuming 4% interest rate, $A^{1}_{30:\overline{10}|}$.
2. `lins2`: whole life insurance for a policyholder aged 30 with benefit payable at the end of month of death at 4% interest rate, $A^{(12)}_{30}$.
3. `lins3`: whole life insurance for a policyholder aged 40 assuming 4% interest rate, A_{40}.
4. `lins4`: 5 years deferred 10-years term life insurance for a policyholder aged 40 assuming 5% interest rate, ${}_{5|10}\bar{A}_{40}$.
5. `lins5`: 5 years annually decreasing term life insurance for a policyholder aged 50 assuming 6% interest rate, $(DA)^{1}_{50:\overline{5}|}$.
6. `lins6`: 10 years increasing term life insurance, age 40, $(IA)^{1}_{40:\overline{10}|}$.

```
R> lins1=Axn(soa08Act, 30,10,i=0.04)
R> lins2=Axn(soa08Act, x=30,i=0.04,k=12)
R> lins3=Axn(soa08Act, 40,i=0.04)
R> lins4=Axn(soa08Act, x=40,n=10,m=5,i=0.05)
R> lins5=DAxn(soa08Act, 50,5)
R> lins6=IAxn(soa08Act, 40,10)
R> c(lins1,lins2,lins3,lins4,lins5,lins6)

[1] 0.01577283 0.20042950 0.27344967 0.03298309 0.08575918 0.15514562
```

Any APV depends by several parameters: the class of insurance benefit, the policyholder's age, the duration of coverage and the interest rate are some of them. Following lines show an interest rate sensitivity analysis on the APV of a ${}_{25}E_{30}$ pure endowment insurance.

```
R> puEnd1<-Exn(soa08Act, x=30, n=35, i=0.06)
R> puEnd2<-Exn(soa08Act, x=30, n=35, i=0.03)
R> c(puEnd1,puEnd2)

[1] 0.1031648 0.2817954
```

Life annuities examples

Life contingencies annuities consist in sequences of payments whose occurrence and duration depend on $\tilde{K}_x$ (or $\tilde{T}_x$). Different types of annuities exist, of which a selection of examples follows. The SOA life table and a policyholder age of 65 apply in all following examples.

1. `annEx1`: annuity immediate, a_{65}.
2. `annEx2`: annuity due, $\ddot{a}_{65}$.
3. `annEx3`: \$ 1,000 annuity due with monthly payment provision, $\ddot{a}_{65}^{(12)}$.
4. `annEx4`: \$ 1,000 annuity due with monthly payment provision 20 years term, $\ddot{a}_{65:\overline{20}|}^{(12)}$.
5. `annEx5`: \$ 1,000 annuity immediate with monthly payment provision 20 years term, $a_{65:\overline{20}|}^{(12)}$.

```
R> annEx1<-axn(soa08Act, x=65, m=1)
R> annEx2<-axn(soa08Act, x=65)
R> annEx3<-12*1000*axn(soa08Act, x=65,k=12)
R> annEx4<-12*1000*axn(soa08Act, x=65,k=12, n=20)
R> annEx5<-12*1000*axn(soa08Act, x=65,k=12,n=20,m=1/12)
R> c(annEx1,annEx2,annEx3,annEx4,annEx5)

[1] 8.896928e+00 9.896928e+00 1.131791e+05 1.082235e+05 1.073211e+05
```

Benefit premiums examples

lifecontingencies package functions can be used to evaluate benefit premium, P, for life contingencies insurance. A (level) benefit premium is defined as the actuarial present value of the provided coverage paid in h installments, $P = \frac{APV}{\ddot{a}_{x:\overline{h}|}}$. The following example displays yearly, `Pa`, and monthly, `Pm`, level benefit premium calculations for a \$ 250,000 35 term life insurance for a 30 years old policyholder, assuming the payment of premium to occur during the first 15 years.

```
R> APV=100000*Axn(soa08Act, x=30,n=35,i=0.025)
R> Pa=APV/axn(soa08Act, x=30,n=15,i=0.025)
R> Pm=APV/(12*axn(soa08Act, x=30,n=15,i=0.025,k=12))
R> c(Pa,Pm)

[1] 921.52623  77.74863
```

Benefit reserves examples

The (prospective) benefit reserve consists in the difference between the APV of future insurers' benefits payments obligations and the APV of projected inflows (remaining scheduled

premiums). It represents the outstanding insurer's obligation to the policyholder for the underwritten insurance policy. An example will better exemplify this concept.
We will evaluate the benefit reserve for a 25 years old 40 years duration life insurance of $ 100,000, with benefits payable at the end of year of death, with level benefit premium payable at the beginning of each year. Assume 3% of interest rate and SOA life table to apply.
The benefit premium and reserve equations for this life contingency insurance are displayed in Equation 11.

$$\begin{aligned} P\ddot{a}_{25:\overline{40}|} &= 100000A^{1}_{25:\overline{40}|} \\ {}_kV^{1}_{25+t:\overline{n-t}|} &= 100000A^{1}_{25+t:\overline{40-t}|} - P\ddot{a}_{25+t:\overline{40-t}|} \end{aligned} \tag{11}$$

```
R> P=100000*Axn(soa08Act,x=25,n=40,i=0.03)/axn(soa08Act,x=25,n=40,i=0.03)
R> reserveFun=function(t) return(100000*Axn(soa08Act,x=25+t,n=40-t,i=0.03)-P*
+ F                                        axn(soa08Act,x=25+t,n=40-t,i=0.03))
R> for(t in 0:40) {if(t%%5==0) cat("At time ",t,
+ F                                " benefit reserve is ", reserveFun(t),"\n")}

At time  0  benefit reserve is  0
At time  5  benefit reserve is  1575.179
At time  10  benefit reserve is  3221.986
At time  15  benefit reserve is  4848.873
At time  20  benefit reserve is  6290.505
At time  25  benefit reserve is  7258.187
At time  30  benefit reserve is  7250.61
At time  35  benefit reserve is  5380.243
At time  40  benefit reserve is  0
```

The calculation of the benefit reserve for a deferred annuity due is the final example of this section. We assume policyholder's age to be 25 and the annuity to be deferred at 65. The reserve equation is ${}_{n|}\ddot{a}_x - \bar{P}({}_{n|}\bar{a}_x)\ddot{a}_{x+k:\overline{n-k}|}$ when $x \ldots n$, $\ddot{a}_{x+k}$ otherwise. The code below calculates the level premium in the initial part and the reserve function while Figure 3 displays the reserve function.

```
R> yearlyRate=12000
R> irate=0.02
R> APV=yearlyRate*axn(soa08Act, x=25, i=irate,m=65-25,k=12)
R> levelPremium=APV/axn(soa08Act, x=25,n=65-25,k=12)
R> annuityReserve<-function(t) {
+ F         out<-NULL
+ F         if(t<65-25) out=yearlyRate*axn(soa08Act, x=25+t,
+ F     i=irate,m=65-(25+t),k=12)-levelPremium*axn(soa08Act,
+ F               x=25+t,n=65-(25+t),k=12) else {
+ F                 out=yearlyRate*axn(soa08Act, x=25+t, i=irate,k=12)
+ F         }
+ F         return(out)
+ F }
```

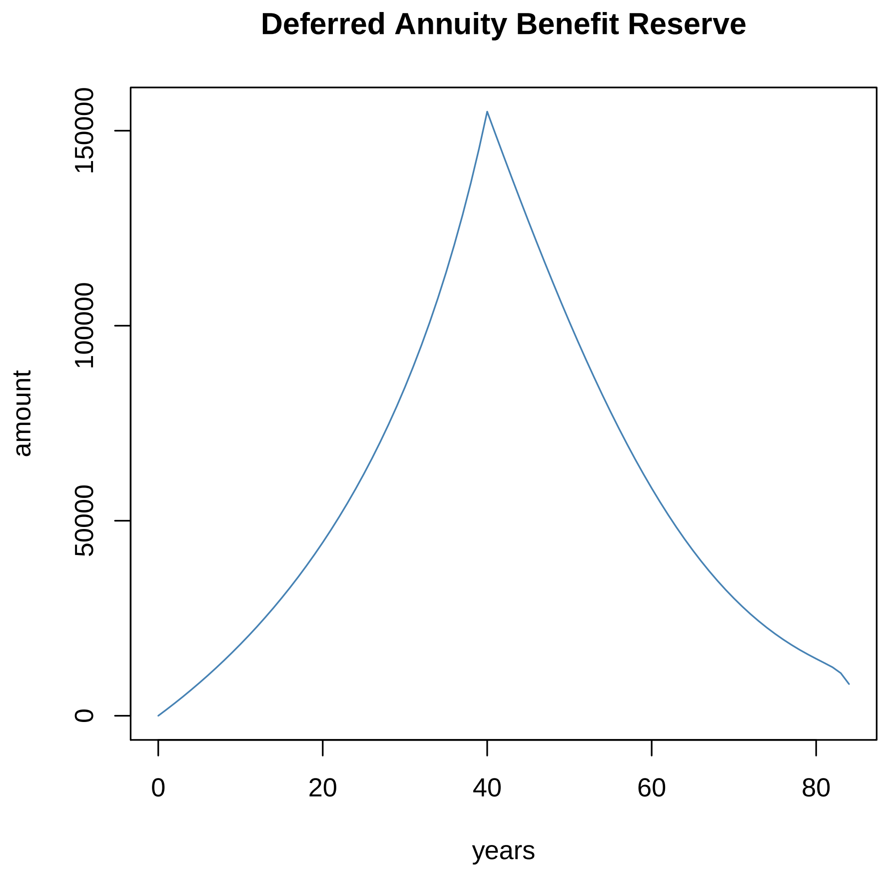

Figure 3: Benefit reserve for $\ddot{a}_{65}$.

```
R> years=seq(from=0, to=getOmega(soa08Act)-25-1,by=1)
R> annuityRes=numeric(length(years))
R> for(i in years) annuityRes[i+1]=annuityReserve(i)
R> dataAnnuityRes<-data.frame(years=years, reserve=annuityRes)
```

Expenses considerations

The premium the policyholder is usually charged to contains an allowance for expenses and profit loading. Those expenses cover the policy servicing, the producers' commission. In some case the insurer profit load is explicitly taken into account in the benefit premium as a flat amount or as a percentage of final premium. In other cases an implicit profit loading is generated by using demographic and financial assumptions more prudential than would be necessary when pricing and reserving the policy. The equivalence principle can be extended to the gross premium, G, and expense augmented reserve, ${}_tV^E$, considering expenses allowance by using Equation 12

$$
\begin{aligned}
G &= APV\left(Benefits\right) + APV\left(Expenses\right) \\
{}_tV^E &= APV\left(Benefits\right) + APV\left(Expenses\right) - APV\left(GrossPremium\right)
\end{aligned}
\tag{12}
$$

The following example shows how to a expense loaded premium G for a \$ 100,000 whole life insurance on a 35 year old insured $100,000A_{35}$ is calculated assuming the following: 10% of premium expense per year, 25 per year of policy expense, annual maintenance expense of 2.5 per 1,000 unit of capital.

The equation to be solved is $G\ddot{a}_{35} = 100000A_{35} + \left(2.5 * 100000/1000 + 25 + 0.1G\right)\ddot{a}_{35}$.

```
R> G=(100000*Axn(soa08Act, x=35)+ 275)/(1-.1)
R> G

[1] 14607.71
```

Insurances and annuities on two lives

lifecontingencies package provides functions designed to evaluate life insurance and annuities on two lives. Following examples check the equality $a_{\overline{xy}} = a_x + a_y - a_{xy}$.

```
R> twoLifeTables=list(maleTable=soa08Act, femaleTable=soa08Act)
R> ex1<-axn(soa08Act, x=65,m=1)+axn(soa08Act, x=70,m=1)-
+ F                  axyn(soa08Act,soa08Act,        x=65,y=70,status="joint",m=1)
R> ex2<-axyzn(twoLifeTables, x=c(65,y=70), status="last",m=1)
R> round(ex1-ex2,2)

[1] 0
```

Reversionary annuities (annuities payable to life y upon death of x), $a_{x|y} = a_y - a_{xy}$ can also be evaluate combining **lifecontingencies** functions.

```
R> axn(soa08Act, x=60,m=1)-axyzn(twoLifeTables, x=c(65,60),status="joint",m=1)

[1] 2.695232
```

4.4. Stochastic analysis

This last section illustrates some stochastic analysis that can be performed by our package, both in demographic analysis and life insurance evaluation. Section 4.4.1 applies stochastic analysis on demographic issues, while Section 4.4.2 applies stochastic analysis on insurance pricing.

Demographic examples

The age-until-death, both in the continuous, $\tilde{T}_x$, or curtate form, $\tilde{K}_x$, is a stochastic variable whose distribution is intrinsic in the deaths within a life table. The code below shows how `rLife` function can be used to draw sample of size 10 from the continuous and curtate future lifetime implicit in the SOA life table.

```
R> sample1<-rLife(n=10,object=soa08Act,x=0,type="Tx")
R> sample2<-rLife(n=10,object=soa08Act,x=0,type="Kx")
```

Next example shows how the mean of the sampled distribution from the curtate future lifetime for a 29 year old policyholder, $\tilde{K}_{29}$, is statistically equal to the expected life time, e_x when `rLife` function is used.

```
R> exn(soa08Act, x=29,type="curtate")

[1] 45.50066

R> t.test(x=rLife(2000,soa08Act, x=29,type="Kx"),
+ F                  mu=exn(soa08Act, x=29,type="curtate"))$p.value

[1] 0.3745385

R> deathsIPS55M<-rLife(n=numSim,ips55M, x=0, type="Kx")
```

Finally, Figure 4 shows the deaths distribution implicit in the ips55M life table generated with the aid of `rLife` function.

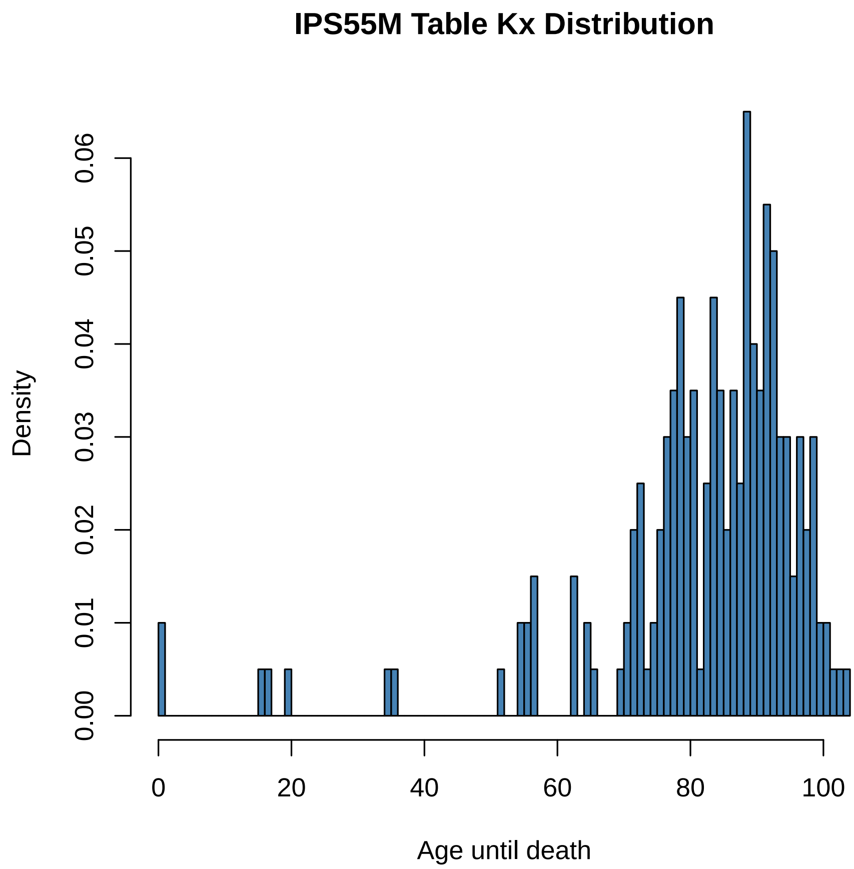

Figure 4: IPS55 deaths distribution function.

Actuarial mathematics examples

The APV is the present value of a random variable, $\tilde{Z}$. $\tilde{Z}$ represents a composite function between the discount amount and indicator variables regarding the life status of the insured. We call $\tilde{Z}$ the present value of benefits random variable, $\tilde{Z}$.
Life contingencies evaluation functions return the APV as default value, since the `type` parameter has "EV" (expected value) as default value. However most life contingencies actuarial mathematics functions are provided with a "ST" (stochastic) argument for `type` parameter. The "ST" argument allows to obtain a sample of size one from the underlying $\tilde{Z}$ distribution. However, when samples of greater dimension are required, the most straightforward approach is to use the `rLifeContingencies` function.
Code below will show $\tilde{Z}$ variate generation from term life insurances, increasing life term insurances, temporary annuity,and endowment insurances respectively. For each example, the unbiaseness is verified by comparing the mean of simulated variate with the theoretical APV using a classical t - test. All examples are referred to an individual aged 20 years old for an insurance duration of 40 years. Figure 5 shows the resulting $\tilde{Z}$ distributions.

```
R> APVAxn=Axn(soa08Act,x=25,n=40,type="EV")
R> APVAxn

[1] 0.0479709

R> sampleAxn=rLifeContingencies(n=numSim, lifecontingency="Axn",
+ F                 object=soa08Act,x=25,t=40,parallel=TRUE)
R> tt1<-t.test(x=sampleAxn,mu=APVAxn)$p.value
R> APVIAxn=IAxn(soa08Act,x=25,n=40,type="EV")
R> APVIAxn

[1] 1.045507

R> sampleIAxn=rLifeContingencies(n=numSim, lifecontingency="IAxn",
+ F                 object=soa08Act,x=25,t=40,parallel=TRUE)
R> tt2<-t.test(x=sampleIAxn,mu=APVIAxn)$p.value
R> APVaxn=axn(soa08Act,x=25,n=40,type="EV")
R> APVaxn

[1] 15.46631

R> sampleaxn=rLifeContingencies(n=numSim, lifecontingency="axn",
+ F                 object=soa08Act,x=25,t=40,parallel=TRUE)
R> tt3<-t.test(x=sampleaxn,mu=APVaxn)$p.value
R> APVAExn=AExn(soa08Act,x=25,n=40,type="EV")
R> APVAExn

[1] 0.1245488

R> sampleAExn=rLifeContingencies(n=numSim, lifecontingency="AExn",
+ F                 object=soa08Act,x=25,t=40,parallel=TRUE)
R> tt4<-t.test(x=sampleAExn,mu=APVAExn)$p.value
R> c(tt1, tt2,tt3, tt4)

[1] 0.6595602 0.9740868 0.7142625 0.4101805
```

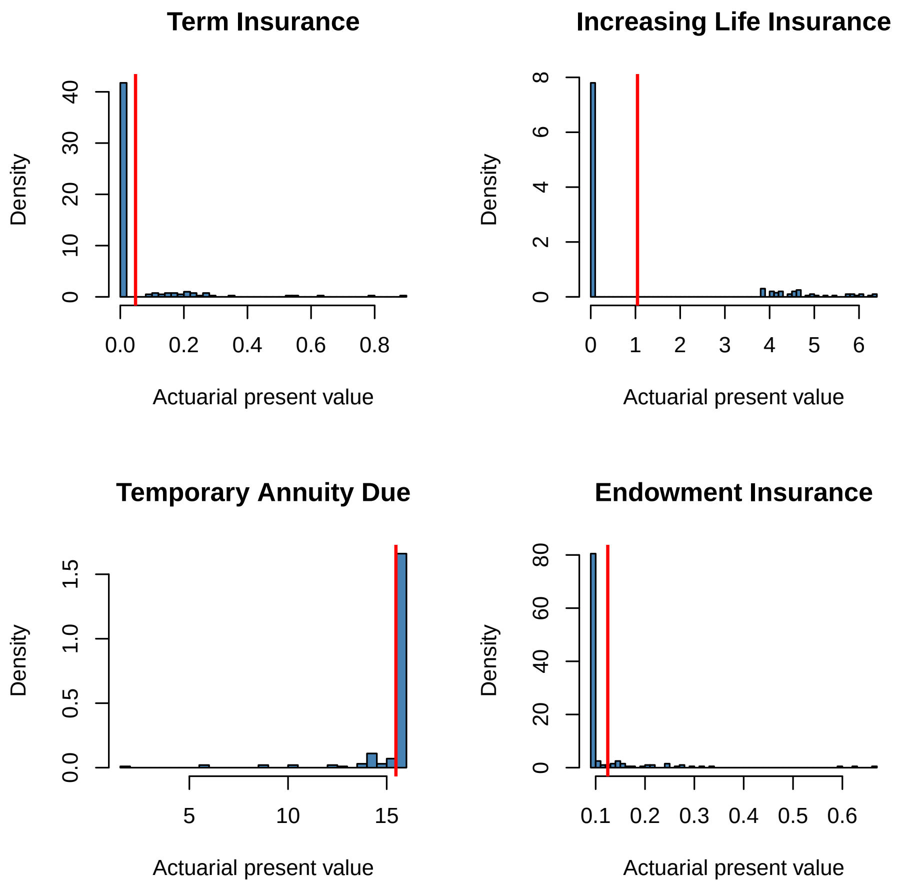

Figure 5: Life insurance stochastic variables distributions. Red vertical line represents APV.

The final example shows how the stochastic functions bundled in **lifecontingencies** can be used to make an actuarial appraisal of embedded benefits as following example shows. Suppose a corporation grants its employees a life insurance benefit equal to the annual salary, payable at the month of death. Suppose moreover that:

1. The expected value and the standard deviation of the salary are \$ 50,000 and \$ 15,000 respectively and salary distribution follows a log-normal distribution.

2. The employees distribution is uniform in the range 25 - 65. Assume 65 to be retirement age.

3. The SOA illustrative table represents an unbiased description of the population mortality.

4. Assume no lapse to hold.

5. The policy length is annual.

We evaluated the best estimate, that is the fair value of the insured benefits according to IAS 19 accounting standards (another word for benefit premium), and a risk margin measure. As risk margin measure we are using the difference between the 75th percentile and the best estimate. IFRS standards, Post, Grandl, Schmidl, and Dorfman (2007), define the fair value of an insurance liability as the sum of its best estimate plus its risk margin.

In the initial part of the example, we set out the parameter of the model and configure the parallel computation facility available by the package **parallel**. The code parallelization has been adapted from examples found in McCallum and Weston (2011) textbook.

```
R> nsim=100
R> employees=500
R> salaryDistribution=rlnorm(n=employees,m=10.77668944,s=0.086177696)
R> ageDistribution=round(runif(n=employees,min=25, max=65))
R> policyLength=sapply(65-ageDistribution, min,1)
R> getEmployeeBenefit<-function(index,type="EV") {
+ F         out=numeric(1)
+ F         out=salaryDistribution[index]*Axn(actuarialtable=soa08Act,
+ F                         x=ageDistribution[index],n=policyLength[index],
+ F                         i=0.02,m=0,k=1, type=type)
+ F         return(out)
+ F }
R> require(parallel)
R> cl <- makeCluster(detectCores())
R> worker.init <- function(packages) {
+ F         for (p in packages) {
+ F                 library(p, character.only=TRUE)
+ F         }
+ F         invisible(NULL)
+ F }
```

```
R> clusterCall(cl,
+ F                 worker.init, c('lifecontingencies'))

[[1]]
NULL

[[2]]
NULL

R> clusterExport(cl, varlist=c("employees","getEmployeeBenefit",
+ F                                "salaryDistribution","policyLength",
+ F                                "ageDistribution","soa08Act"))
```

Then we perform best estimate and risk margin calculations.

```
R> employeeBenefits=numeric(employees)
R> employeeBenefits<- parSapply(cl, 1:employees,getEmployeeBenefit, type="EV")
R> employeeBenefit=sum(employeeBenefits)
R> benefitDistribution=numeric(nsim)
R> yearlyBenefitSimulate<-function(i)
+ F {
+ F         out=numeric(1)
+ F         expenseSimulation=numeric(employees)
+ F         expenseSimulation=sapply(1:employees, getEmployeeBenefit, type="ST")
+ F         out=sum(expenseSimulation)
+ F         return(out)
+ F }
R> benefitDistribution <- parSapply(cl, 1:nsim,yearlyBenefitSimulate )
R> stopCluster(cl)
R> riskMargin=as.numeric(quantile(benefitDistribution,.75)-employeeBenefit)
R> totalBookedCost=employeeBenefit+riskMargin
R> employeeBenefit

[1] 128586.7

R> riskMargin

[1] 49497.97

R> totalBookedCost

[1] 178084.7
```

5. Discussion

5.1. Advantages and limitations

The **lifecontingencies** package allows actuaries to perform demographic, financial and actuarial mathematics calculations within R software. Pricing, reserving and stochastic evaluations of life insurance contract can be therefore performed using R. Moreover, an original feature of **lifecontingencies** is the ability to generate samples variate from both life tables and life insurances stochastic distributions.

One of the most important limitations of **lifecontingencies** is that it handles only single decrements tables. Another limitation is that currently it does not allow continuous time life contingencies to be modeled.
We expect to remove such limitations in the future. Similarly, we expect to to provide coerce methods toward packages specialized in demographic analysis, like **demography** and **LifeTables** packages. Communication with interest rates modelling packages, as **termstrcR** will be also explored.

5.2. Accuracy

The accuracy of calculation have been verified by checking with numerical examples reported in Bowers and of Actuaries (1986) and in the lecture notes of Actuarial Mathematics the author attended years ago at Catholic University of Milan, Mazzoleni (2000). The numerical results are identical to those reported in the Bowers and of Actuaries (1986) textbook for most function, with the exception of fractional payments annuities where the accuracy leads only to the 5th decimal. The reason of such inaccuracy is due to the fact that the package calculates the APV by directly sum of fractional survival probabilities, while the formulas reported in Bowers and of Actuaries (1986) textbook uses an analytical formula.

Finally, it is worth to remember that the package and functions herein are provided as is, without any guarantee regarding the accuracy of calculations. The author disclaims any liability arising by eventual losses due to direct or indirect use of this package.

Acknowledgments

The author wishes to thank all those whose suggestions contributed to the package enhancements. A special thank however is for Christophe Dutang, for his suggestions and vignette revision, and Tim Riffle.

References

Anderson J (1999). *Commutation Functions*. Education and Examination Committee of the Society of Actuaries.

Booth H, Hyndman R, Tickle L, De Jong P (2006). "Lee-Carter Mortality Forecasting: A Multi-Country Comparison of Variants and Extensions." *Demographic Research*, **15**(9), 289–310. doi:10.4054/DemRes.2006.15.9.

Bowers N, of Actuaries S (1986). *Actuarial mathematics.* Society of Actuaries. ISBN 9780938959106.

Broverman S (2008). *Mathematics of Investment and Credit.* ACTEX Academic Series. ACTEX Publications. ISBN 9781566986571.

Chambers J (2008). *Software for Data Analysis: Programming with R.* Statistics and computing. Springer. ISBN 9780387759357.

Chris Ruckman, Joe Francis (2006). *Financial Mathematics: A Practical Guide for Actuaries and Other Business Professionals.* Warren Centre for Actuarial Studies and Research, second edition edition.

Christophe Dutang, Vincent Goulet, Mathieu Pigeon (2008). "**actuar**: An R Package for Actuarial Science." *Journal of Statistical Software*, **25**(7), 38. URL http://www.jstatsoft.org/v25/i07.

Ferstl R, Hayden J (2010). "Zero-Coupon Yield Curve Estimation with the Package **termstrc**." *Journal of Statistical Software*, **36**(1), 1–34. URL http://www.jstatsoft.org/v36/i01/.

Gesmann M, Zhang Y (2011). ***ChainLadder**: Mack, Bootstrap, Munich and Multivariate-chain-ladder Methods.* R package version 0.1.4-3.4.

Guirreri S (2010). *Simulating the Term Structure of Interest Rates with Arbitrary Marginals.* Ph.D. thesis, University of Palermo - Department of Statistics and Mathematics "S. Vianelli", Palermo. URL http://www.guirreri.host22.com.

Keyfitz N, Caswell H (2005). *Applied Mathematical Demography.* Statistics for biology and health. Springer. ISBN 9780387225371.

Klugman S, Panjer H, Willmot G, Venter G (2009). *Loss Models: From Data to Decisions.* Third edition. Wiley New York.

MATLAB (2010). *Version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts.

Mazzoleni P (2000). *Appunti di Matematica Attuariale.* EDUCatt Università Cattolica. ISBN 9788883110825.

McCallum Q, Weston S (2011). *Parallel R.* O'Reilly Media. ISBN 9781449309923.

Norusis M (2008). *SPSS 16.0 Guide to Data Analysis.* 2nd edition. Prentice Hall Press, Upper Saddle River, NJ, USA. ISBN 0136061362, 9780136061366.

Post T, Grandl H, Schmidl L, Dorfman MS (2007). "Implications of IFRS for the European Insurance Industry Insights from Capital Market Theory." *Risk Management and Insurance Review*, **10**(2), 247–265. ISSN 1540-6296. doi:10.1111/j.1540-6296.2007.00117.x. URL http://dx.doi.org/10.1111/j.1540-6296.2007.00117.x.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Riffe T (2011). *LifeTable: **LifeTable**, a Package with a Small Set of Useful Lifetable Functions.* R package version 1.0.1, URL http://sites.google.com/site/timriffepersonal/r-code/lifeable.

Rigby RA, Stasinopoulos DM (2005). "Generalized Additive Models for Location, Scale and Shape." *Applied Statistics*, **54**, 507–554.

Rob J Hyndman, Heather Booth, Leonie Tickle, John Maindonald (2011). ***demography**: Forecasting Mortality, Fertility, Migration and Population Data.* R package version 1.09-1, URL http://CRAN.R-project.org/package=demography.

SAS Institute Inc (2003). *SAS/STAT Software, Version 9.1.* Cary, NC. URL http://www.sas.com/.

Zhang W (2011). ***cplm**: Monte Carlo EM Algorithms and Bayesian Methods for Fitting Tweedie Compound Poisson Linear Models.* R package version 0.2-1, URL http://CRAN.R-project.org/package=cplm.

Affiliation:

Giorgio Alfredo Spedicato
StatisticalAdvisor
Via Firenze 11 20037 Italy
Telephone: +39/334/6634384
E-mail: lifecontingencies@statisticaladvisor.com
URL: www.statisticaladvisor.com