Introduction to lifecontingencies Package

Giorgio Alfredo Spedicato, Ph.D

Abstract

lifecontingencies 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 new insurance products and to determine portolios' risk based capital requirements.

This paper briefly summarizes the theory regarding life contingencies, that is represented by concepts of financial and actuarial mathematics. Then it shows how **lifecontingencies** functions represent a perfect cookbook to perform life insurance actuarial analysis and related stochastic simulations throught applied examples.

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

1. Introduction

As of March 2012, **lifecontingencies** appears to be the first R package that deals with life insurance evaluation. 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**, Dutang, Goulet, and 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. GLM models, 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, e.g., GAMLSS and Tweedie regressions, can be fit using specifically developed packages as **gamlss**, Rigby and Stasinopoulos (2005), and **cplm**, Zhang (2011), packages respectively.
Life insurance evaluation models demographic and financial data, mainly . R has a dedicated view to 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 exists that performs life contingencies calculations, as of March 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 softwares 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 book, Bowers, Gerber, Hickman, Jones, and Nesbitt (1997). Since life contingencies theory grounds on demography and classical financial mathematics, w have made an exstensive use of Ruckman and 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. E.g. the insurance pure premiun (also known as benefit premium) is the present value of the series of future cash flows whose 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 individualsliving 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.

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, $\tilde{K}_x$, i.e., 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 $x+t$.
- ${}_tL_x = \sum_{k=0}^{t} l_{x+k}$, 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, $e_x = E\left(K_x\right)$ and its complete form $\mathop{\overset{\circ}{e}}_{x} = E\left(T_x\right)$.

The Keyfitz textbook, Keyfitz and Caswell (2005), provides an exaustive 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 Formula 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_t, represents the measure of the price of money available in future times. This paper will use i to express the effective (real) compound interest. It means that if i is the interest rate, a sum of 1 monetary unit accumulates throught time according to the accumulation function, $A\left(t\right) = \left(1+i\right)^t$. Arrangements lead to discount and nominal (m-compound) interest rates as shown in Formula 1.

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

All financial mathematics functions (such annuities, $\bar{a}_{\overline{n}|}$, or accumulated values, $s_{\overline{n}|}$) can be written as a particular case of formula 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. They consist in present values whose amounts are not certain, since both the time of their (eventual) occurrence and their final values depend by events regarding the life of the policyholder (that is the reason for which they are called life contingencies). **lifecontingencies** package provides function 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 premim. The benefit premiums plus the loadings 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 *et al.* (1997) 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 perfomed at the end of year of death, we can write $\tilde{Z}$ as $\tilde{Z} = \begin{cases} b * v^{\tilde{K}_x+1}, \tilde{K}_x \leq n \\ 0, \tilde{K}_x > n \end{cases}$ The APV symbol 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. Assuming \$1 payment, we can write $\tilde{Z}$ as $\tilde{Z} = a_{\overline{\tilde{K}+1}|}$. 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 $v^n * I\left(\tilde{K}_x \geq n\right)$ and its APV expression is ${}_nE_x$.

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 $\tilde{Z} = v^{\min\left(n, \tilde{K}_x\right)}$, while its APV symbol is $A_{x:\overline{n}|}^{\ 1}$.

We send interested readers to the Bowers *et al.* (1997) textbook for formulas regarding other life contingencies insurances as $(DA)^1_{x:\overline{n}|}$, the decreasing term life insurance, $(IA)_x$, the increasing term life insurance, and common variations on payment form arrangments like deferrment and fractional payments.

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 tecniques.
Commutation tables extend life table by tabulating special function of age and rate of interest whose ratios allow the actuary to evaluate APV for standard insurances. The Anderson (1999) paper provides a comprehensive overview of this topic. 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: $\bar{z} = \left\langle \bar{pk} \bullet \bar{x}\right\rangle$. $\bar{pk}$ is $Pr\left[\tilde{K} = k\right]$, that is the probability that the future (integer) remaining years will be exactly k, $\bar{x}$ 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 Formula 3, while Formula 4 reports the APV using the current payment technique. Finally, Formula 5 calculates the APV using the expected value approach.

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

$$APV = \sum_{k=0}^{\min(\omega - x - 1, n)} p_{x,t} \tag{4}$$

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

3. The structure of the package

Package **lifecontingencies** contains classes and methods to handle lifetables 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> #definition of lifetable
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: demographics functions, financial mathematics functions and actuarial mathematics functions.

The demographic functions group comprises the followings:

1. `dxt` returns deaths between age x and $x+t$, $d_{x,t}$.
2. `pxt` returns survival probability between age x and $x+t$, $p_{x,t}$.
3. `pxyt` returns the survival probability for two lifes, $d_{xy,t}$.
4. `qxt` returns death probability between age x and $x+t$, $q_{x,t}$.
5. `qxyt` returns the survival probability for two lifes, $q_{xy,t}$.
6. `Txt` returns the number of person-years lived after exact age x, $T_{x,t}$.
7. `mxt` returns central mortality rate, $m_{x,t}$.
8. `exn` returns the complete or curtate expectation of life from age x to $x+n$, $e_{x,n}$.
9. `rLife` returns a sample from the time until death distribution underlying a life table.
10. `exyt` returns the expected life time for two lifes between age x and $x+t$.
11. `probs2lifetable` returns a life table l_x from raw one - year survival / death probabilities.

The financial mathematics group comprises the followings:

1. `presentValue` returns the present value for a series of cash flows, $PV = \sum_i CF_i * v^{t_i}$.
2. `annuity` returns the present value of a annuity - certain, $a_{\overline{n}|}$.
3. `accumulatedValue` returns the future value of a series of cash flows, $s_{\overline{n}|}$.
4. `increasingAnnuity` returns the present value of an increasing annuity - certain, $(IA)_n$.
5. `decreasingAnnuity` returns the present value of a decreasing annuity, $(DA)_{\overline{n}|}$.
6. `nominal2Real` returns the effective annual interest (discount) rate i given the nominal m-periodal interest $i^(k)$ or discount $d^(k)$ rate.
7. `real2Nominal` returns the m-periodal interest or discount rate given the m periods or the discount.
8. `intensity2Interest` returns the intensity of interest δ given the interest rate i.
9. `interest2Intensity` returns the interest rate i given the intensity of interest δ.
10. `duration` returns the duration of a series of cash flows, $\sum_t^T \frac{t*CF_t\left(1+\frac{i}{m}\right)^{-t*m}}{P}$.
11. `convexity` returns the convexity of a series of cash flows, $\sum_t^T t*\left(t+\frac{1}{m}\right)*CF_t\left(1+\frac{y}{m}\right)^{-m*t-2}$.

The actuarial mathematics group comprises the following functions, for which we report must important function:

1. `Axn` models one head life insurance, whose APV symbol is $A^1_{x:\overline{n}|}$.
2. `AExn` models the n-year term insurance, whose APV symbol is $A_{x:\overline{n}|}^{\;\;\;1}$.
3. `Axyn` models two heads life insurances, whose APV symbol is $\bar{A}^1_{\overline{xy}:\overline{n}|}$.
4. `axn` models annuities, whose APV symbol is $\ddot{a}_x$.
5. `axyn` models two heads annuities, whose APV symbol is $\ddot{a}_{xy}$.
6. `Exn` models pure endowment, whose APV symbol is ${}_nE_x$.
7. `Iaxn` models the increasing annuity, whose APV symbol is $(Ia)_x$.
8. `IAxn` models the increasing life insurance, whose APV symbol is $(IA)_x$.
9. `DAxn` models the decreasing life insurance, whose APV symbol is $(DA)_x$.

As general remark, standard financial and actuarial mathematics functions parameters are:

- `x`, the policyholder's age at the policy issuance time.
- `n`, the coverage duration that could be missing if the policy lasts for the remaining lifetime. For financial mathematics function it represent the lenght of the payment.
- `actuarialtable`, an actuarial table on which life insurance calculation are performed.
- `i`, the interest rate, that in some case could be time - varying.
- `k`, the frequency of payments per year (default value is 1).

4. Code and examples

4.1. Classical financial mathematics example

The **lifecontingencies** package provides functions to perform classical financial mathematics calculations. Following examples 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

The code below shows how to switch from effective interest rates (APR) to nominal interest rates, $i^{(m)} \to i$, and vice versa. Similarly, it is possible to work with discount rates also, that is $d^{(m)} \to d$ and vice versa.

```
R> #an APR of 3% is equal to a
R> real2Nominal(0.03,12)

[1] 0.02959524

R> #nominal interest rate
R> #while 6% annual nominal interest rate is the same of
R> nominal2Real(0.06,12)

[1] 0.06167781

R> #4% effective interest rate corresponds to
R> real2Nominal(0.04,4)*100

[1] 3.941363

R> #nominal interest rate (in 100s) compounded quarterly
R>
R> #an effective rate of discount of 4% is equal to a
R> real2Nominal(i=0.04,k=12,type="discount")

[1] 0.04075264

R> #nominal rate of discount payable quarterly
```

Present value analysis

Performing a project appraisal means evaluating the net present value (NPV) of all projected cash flows. Following code examples will show NPV evaluation assuming a varying cash flow pattern (example 1), varying interest rate (example 2) and uncertain cash flow (example 3).

```
R> #varing cash flow pattern
R> capitals=c(-1000,200,500,700)
R> times=c(0,1,2,5)
R> ex1<-presentValue(cashFlows=capitals, timeIds=times,
+ F              interestRates=0.03)
R> #varying interest rates
R> ex2<-presentValue(cashFlows=capitals, timeIds=times,
+ F              interestRates=c( 0.04, 0.02, 0.03, 0.05))
R> #uncertain cash flows
R> ex3<-presentValue(cashFlows=capitals, timeIds=times,
+ F interestRates=c( 0.04, 0.02, 0.03, 0.05), probabilities=c(1,1,1,0.5))
R> c(ex1,ex2,ex3)

[1] 269.29886 215.84470 -58.38946
```

Annuities and future values

Code below shows examples of annuities ($a_{\overline{n}|}$) and accumulated values ($s_{\overline{n}|}$) evaluations:

```
R> #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

R> #while the corresponding future value is
R> 100*accumulatedValue(i=0.03,n=5)

[1] 530.9136
```

A more concrete and meaningful example follows. A man wants to save \$ 100,000 to pay for the education of his son in 10 years time. An education fund requires the investors to deposit equal instalments annually at the end of each year. If interest of 0.05 is paid, how much does the man need to save each year (R) in order to meet his target?

```
R> C=100000
R> R=C/accumulatedValue(i=0.05,n=10)
R> R

[1] 7950.457
```

The fractional payments annuities represent an important class of financial contract. An annuity with m fractional payments per period grants a payment of $\frac{1}{m}$ during each period. **lifecontingencies** package allows fractional annuities ($a^{(m)}{}_{\overline{n}|}$) can be handled when using `annuity` and `accumulatedValue` functions. The present value of an annuity-immediate of 100 per quarter for 4 years, assuming interest to be compounded semiannually at the nominal rate of 6% is

```
R> 100*4*annuity(i=nominal2Real(0.06,2),n=4,k=4)
```

```
[1] 1414.39
```

`increasingAnnuity` and `decreasingAnnuity` functions handle increasing and decreasing annuities, whose symbols are $(IA)_x$, $(DA)_x$. Code below exemplifies these function, assuming a ten years duration and a 3% interest rate.

```
R> #increasing annuity example
R> ex1<-increasingAnnuity(i=0.03, n=10,type="due")
R> #decreasing annuity example
R> ex2<-decreasingAnnuity(i=0.03, n=10,type="immediate")
R> c(ex1,ex2)
```

```
[1] 46.18416 48.99324
```

The last example of this section exemplifies the calculation of the present value of a geometrically increasing annuity. We will assume each year the annuity increases its value by 3%, being the interest rate is 4% and the annuity duration being 10 years.

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

```
[1] 9.48612
```

Loan amortization

lifecontingencies financial mathematics function allow to define the repayments schedule of any loan arrangment, as exemplified in this section. In the following example, let C denote the loaned capital (principal), then assuming an interest rate i, the amount due to the lender at each instalment 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. Figure **??** showS the end of period (EoY) balance due for a 30 - years duration loan, assuming a 5% interest rate on a principal of $ 100,000.

```
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> installment=
+ F 1/payments_per_year*capital/annuity(i=interest, n=years,k=payments_per_year)
R> installment
```

```
[1] 3212.9
```

```
R> #compute the balance due at the beginning of each period
R> balance_due=numeric(years*payments_per_year)
R> balance_due[1]=capital*(1+rate_per_period)-installment
R> for(i in 2:length(balance_due)) balance_due[i]=balance_due[i-1]*(1+rate_per_period)-ins
```

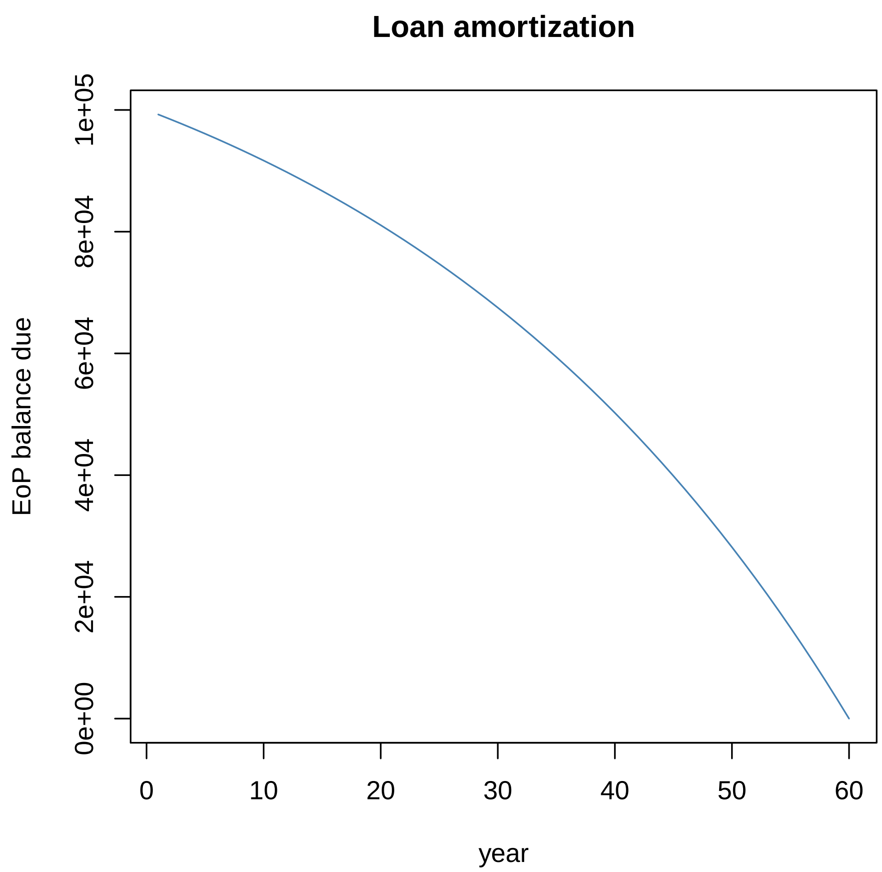

Figure 1: Loan amortization: end of period (EoP) balance due

Bond pricing

Bond pricing represents another application of present value analysis. A standard bond whose principal (face value) will be repaid at time T consists in a sequence of coupons c_t, priced according to a coupon rate $j^{(k)}$ on a principal C. Equation 6 expresses the present value of a bond.

$$B_t = c_t a^{(k)}{}_{\overline{n|}} + Cv^T \tag{6}$$

Examples follow showing how **lifecontingencies** package functions can be used to perform bond pricing.

```
R> #define a function to compute bond market value
R> bond<-function(faceValue, couponRate, couponsPerYear, yield,maturity)
+ F {
+ F         out=NULL
+ F         numberOfCF=maturity*couponsPerYear #determine the number of CF
+ F         CFs=numeric(numberOfCF)
+ F         payments=couponRate*faceValue/couponsPerYear #determine the coupon sum
+ F         cf=payments*rep(1,numberOfCF)
+ F         cf[numberOfCF]=faceValue+payments #set the last payment amount
+ F         times=seq.int(from=1/couponsPerYear, to=maturity, by=maturity/numberOfCF)
+ F         out=presentValue(cashFlows=cf, interestRates=yield, timeIds=times)
+ F         return(out)
+ F }
R> #coupon rate 6%, two coupons per year, face value 1000,
R> #yield 5%, three years to maturity
R> bndEx1<-bond(1000,0.06,2,0.05,3)
R> #coupon rate 3%, one coupons per year,
R> #face value 1000, yield 3%, three years to maturity
R> bndEx2<-bond(1000,0.06,1,0.06,3)
R> c(bndEx1, bndEx2)

[1] 1029.25 1000.00
```

Last financial mathematics examples show how duration and convexity of cash flows can be estimated by **lifecontingencies** package functions.

```
R> #set cash flows, times and interest rates
R> cashFlows=c(100,100,100,600,500,700)
R> timeVector=seq(1:6)
R> interestRate=0.03
R> #dollar duration
R> duration(cashFlows=cashFlows, timeIds=timeVector,
+ F                 i=interestRate, k = 1, macaulay = FALSE)

[1] 4.563124
```

```
R> #Macaulay duration
R> duration(cashFlows=cashFlows, timeIds=timeVector,
+ F                 i=interestRate, k = 1, macaulay = TRUE)

[1] 4.430218

R> #convexity
R> convexity(cashFlows=cashFlows, timeIds=timeVector,
+ F                 i=interestRate, k = 1)

[1] 25.74647
```

4.2. Lifetables 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

Lifetable 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, i.e. 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, lx, px and ex 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, as shown below.

```
R> #head method
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`. It would be also the most real concrete approach an actuary would use to handle `lifetable` objects. In the following example the US Social Security life table for males and females as long as the Italian IPS55 tables series are loaded from the existing `demoUsa` and `demoIta` datasets bundled in the **lifecontingencies** package.

```
R> #load USA Social Security LT
R> data(demoUsa)
R> usaMale07=demoUsa[,c("age", "USSS2007M")]
R> usaMale00=demoUsa[,c("age", "USSS2000M")]
R> #coerce from data.frame to lifecontingencies
R> #requires x and lx names
R> names(usaMale07)=c("x","lx")
R> names(usaMale00)=c("x","lx")
R> #apply coerce methods and changes names
R> usaMale07Lt<-as(usaMale07,"lifetable")
R> usaMale07Lt@name="USA MALES 2007"
R> usaMale00Lt<-as(usaMale00,"lifetable")
R> usaMale00Lt@name="USA MALES 2000"
R> #compare expected lifetimes
R> c(exn(usaMale00Lt,0),exn(usaMale07Lt,0))

[1] 73.52997 74.88162

R> #load Italian IPS55 tables
R> ##males
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> ##females
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> #finalize the tables
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")
R> #compare expected lifetimes
R> c(exn(ips55M,0),exn(ips55F,0))

[1] 84.32625 88.22466
```

The last way a `lifetable` object can be created is from one year survival or death probabilities. This feature is useful when used in conjunction with the results of a mortality projection method (e.g. Lee - Carter).

```
R> #use 2002 Italian males life tables
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> #reconvert in data frame
R> itaM2002<-as(itaM2002Lt,"data.frame")
R> #add qx
R> itaM2002$qx<-1-itaM2002$px
R> #reduce to 20% one year death probability for ages between 20 and 60
R> for(i in 20:60) itaM2002$qx[itaM2002$x==i]=0.2*itaM2002$qx[itaM2002$x==i]
R> #otbain the reduced mortality table
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> #assume 3% interest rate
R> exampleAct=new("actuarialtable",x=exampleLt@x, lx=exampleLt@lx, interest=0.03,
+ F                  name="example actuarialtable")
```

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

```
R> getOmega(exampleAct)
```

```
[1] 9
```

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

```
R> #apply method print applied on a 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

R> #apply method print applied on an actuarial table
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. Figure 2 plots the survival function (i.e. the plot of x vs l_x) of the SOA illustrative life table.

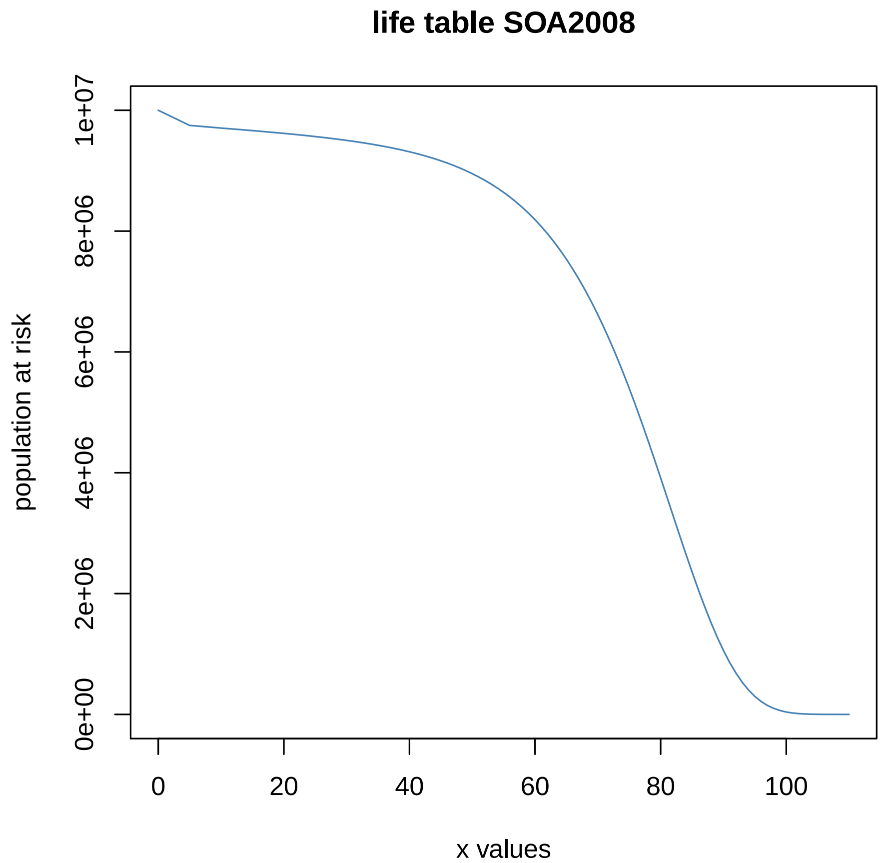

Figure 2: Society of Actuaries illustrative lifetable underlying survival distribution function

Basic demographic analysis

Basic demographic estimations can be performed on valid `lifetable` or `actuariatable` objects. Code below shows how ${}_tp_x$, ${}_tq_x$ and $\mathring{e}_{x:\overline{n}|}$ can be obtained.

```
R> #using ips55M life table
R> #probability to survive one year, being at age 20
R> pxt(ips55M,20,1)

[1] 0.9995951

R> #probability to die within two years, being at age 30
R> qxt(ips55M,30,2)

[1] 0.001332031

R> #expected (curtate) life time between 50 and 70 years
R> exn(ips55M, 50,20)

[1] 19.43322
```

Fractional survival probabilities can also be calculated according with linear interpolation (default value), constant force of mortality and hyperbolic Balducci's assumption, as Bowers *et al.* (1997) details. We will show these concepts on the Society of Actuaries illustrative life table, assuming insured age to be 80 years old.

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

[1] 0.9598496 0.9590094 0.9581701
```

Two heads survival probabilities calculations can be performed also. Code below shows how joint survival probabilities, last survival probabilities and expected joint lifetime can be evaluated using **lifecontingencies** functions.

```
R> jsp=pxyt(ips55M,ips55F,x=65, y=63, t=2) #joint survival probability
R> lsp=pxyt(ips55M,ips55F,x=65, y=63, t=2,status="last") #last survival probability
R> jelt=exyt(ips55M, ips55F, x=65,y=63, status="joint") #joint expected lifetime
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 *et al.* (1997) 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 (i.e. the lump sum benefit premium) for:

1. 10-year term life insurance for a subject aged 30 assuming 4% interest rate, $A^{1}_{30:\overline{10}|}$.
2. 10-year term life insurance for a subject aged 30 with benefit payable at the end of month of death at 4% interest rate,$(A^{1}_{30:\overline{10}|})^{(m)}$.
3. whole life insurance for a subject aged 40 assuming 4% interest rate, A_{40}.
4. 5 years deferred 10-years term life insurance for a subject aged 40 assuming 5% interest rate, ${}_{5|10}\bar{A}_{40}$.
5. 5 years annually decreasing term life insurance for a subject aged 50 assuming 6% interest rate, $(DA)^{1}_{50:\overline{5}|}$.
6. 20 years increasing term life insurance, age 40, $(IA)^{1}_{50:\overline{5}|}$.

```
R> #10 years term life insurance for a 40 years old insurer @ 4% interest
R> lins1=Axn(soa08Act, 30,10,i=0.04)
R> #same as above but payable at the end of month of death
R> lins2=Axn(soa08Act, x=30,n=10,i=0.04,k=12)
R> #whole life variation @6% interest rate (implicit in SOA actuarial table)
R> lins3=Axn(soa08Act, 40)
R> #5-year deferred life insurance, 10 years length, 40 years old, @5% interest rate
R> lins4=Axn(soa08Act, x=40,n=10,m=5,i=0.05)
R> #five years annually decreasing term life insurance, insured aged 50.
R> lins5=DAxn(soa08Act, 50,5)
R> #20 years term annually increasing life insurance, age 40
R> lins6=IAxn(soa08Act, 40,10)
R> c(lins1,lins2,lins3,lins4,lins5,lins6)

[1] 0.01577283 0.01605995 0.16132416 0.03298309 0.08575918 0.15514562
```

Pure endowments APV, ${}_nE_x$, examples are shown by following lines of code:

```
R> #evaluate the APV for a n year pure endowment, age x=30, n=35, i=6%
R> ex1<-Exn(soa08Act, x=30, n=35, i=0.06)
R> #the same but @ i=3%
R> ex2<-Exn(soa08Act, x=30, n=35, i=0.03)
R> c(ex1,ex2)
```

```
[1] 0.1031648 0.2817954
```

Life annuities examples

Following examples show APV calculations for different annuities variations.

```
R> #annuity immediate
R> ex1<-axn(soa08Act, x=65, m=1)
R> #annuity due
R> ex2<-axn(soa08Act, x=65)
R> #due with monthly payments of $1000 provision
R> ex3<-12*1000*axn(soa08Act, x=65,k=12)
R> #due with montly payments of $1000 provision, 20 - years term
R> ex4<-12*1000*axn(soa08Act, x=65,k=12, n=20)
R> #immediate with monthly payments of 1000 provision, 20 - years term
R> ex5<-12*1000*axn(soa08Act, x=65,k=12,n=20,m=1/12)
R> c(ex1,ex2,ex3,ex4,ex5)
```

```
[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, $P = \frac{APV}{\ddot{a}_{x:\overline{h}|}}$.

```
R> #Assume X, aged 30, whishes to buy a $ 250K 35-years life insurance
R> #premium paid annually for 15 years @2.5% interest rate.
R> Pa=100000*Axn(soa08Act, x=30,n=35,i=0.025)/axn(soa08Act, x=30,n=15,i=0.025)
R> #while if the premium is paid on a montly basis the flat benefit premium
R> Pm=100000*Axn(soa08Act, x=30,n=35,i=0.025)/axn(soa08Act, x=30,n=15,i=0.025,k=12)
R> c(Pa,Pm)
```

```
[1] 921.5262 932.9836
```

```
R> #level semiannual premium for an endowment insurance of 10000
R> #insured age 50, insurance term is 20 years
R> APV=10000*(Axn(soa08Act,50,20)+Exn(soa08Act,50,20))
R> P=APV/axn(soa08Act,50,20,k=2)
R> P
```

```
[1] 325.1927
```

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, whith benefits payable at the end of year of death, whith level benefit premium payable at the beginning of each year. Assume 3% of interest rate and SOA life table to apply.
The benefit premium P is determined by equation $P\ddot{a}_{25:\overline{40|}} = 100000 A^{1}_{25:\overline{40|}}$, while the benefit reserve is determined by equation ${}_kV^{1}_{25+t:\overline{n-t|}} = 100000 A^{1}_{25+t:\overline{40-t|}} - P\ddot{a}_{25+t:\overline{40-t|}}$ for $t = 0 \dots 40$.

```
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

R>
R>
```

Figure 3 shows the benefit reserve for a whole life annuity due with level annual premium as ${}_kV({}_{n|}\ddot{a}_x)$. It is equal to ${}_{n|}\ddot{a}_x - \bar{P}({}_{n|}\bar{a}_x)\ddot{a}_{x+k:\overline{n-k|}}$ when $x \dots n$, $\ddot{a}_{x+k}$ otherwise. Figure 3 displays benefit reserve for a 65 years old insured annuity immediate, with 40 years of deferral.

Insurance and annuities on two heads

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

```
R> axn(soa08Act, x=65,m=1)+axn(soa08Act, x=70,m=1)-
+ F                  axyn(soa08Act,soa08Act,        x=65,y=70,status="joint",m=1)

[1] 10.35704

R> axyn(soa08Act,soa08Act, x=65,y=70, status="last",m=1)

[1] 10.35704
```

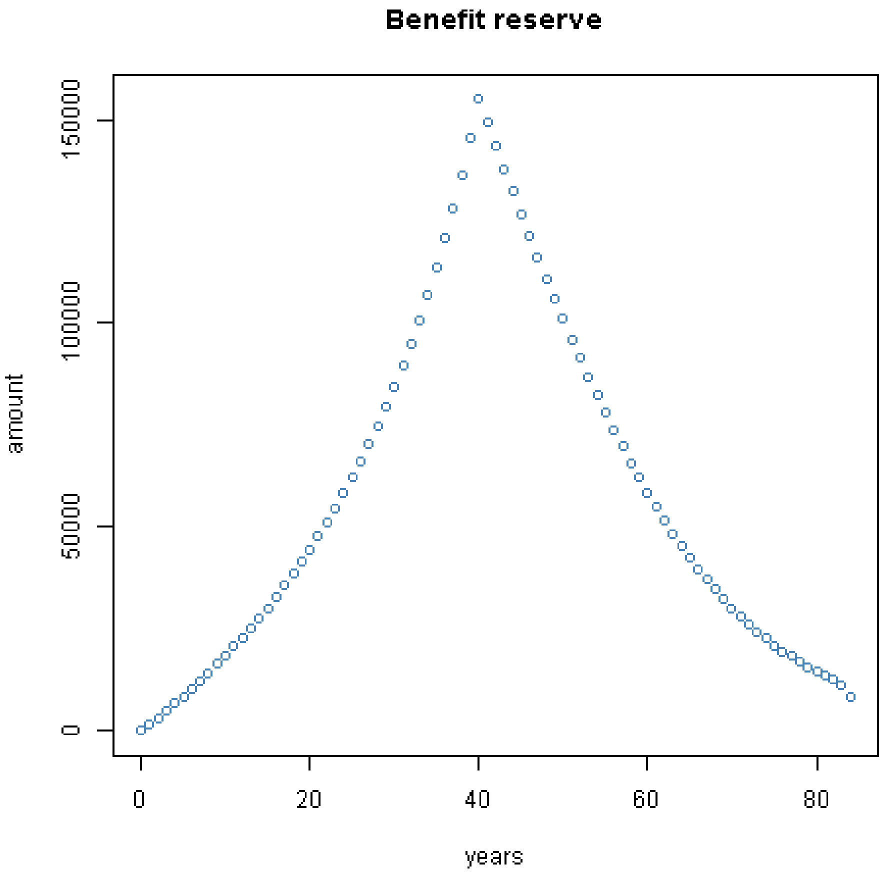

Figure 3: Benefit reserve for $\ddot{a}_{65}$

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> #assume x aged 65, y aged 60
R> axn(soa08Act, x=60,m=1)-axyn(soa08Act,soa08Act, x=65,y=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.

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 to sample values from the age-until-death distribution implicit in the SOA life table.

```
R> data(soa08Act)
R> #sample 10 numbers from the Tx distribution
R> sample1<-rLife(n=10,object=soa08Act,x=0,type="Tx")
R> #sample 10 numbers from the Kx distribution
R> sample1<-rLife(n=10,object=soa08Act,x=0,type="Kx")
```

while code below shows how the mean of the sampled distribution is statistically equivalent to the expected life time, e_x.

```
R> #assume an insured aged 29
R> #his expected integer number of years until death is
R> exn(soa08Act, x=29,type="curtate")

[1] 45.50066

R> #check if we are sampling from a statistically equivalent distribution
R> t.test(x=rLife(2000,soa08Act, x=29,type="Kx"),
+ F                  mu=exn(soa08Act, x=29,type="curtate"))$p.value

[1] 0.7143122

R> #statistically not significant
```

Finally, Figure 4 shows the deaths distribution implicit in the ips55M life table.

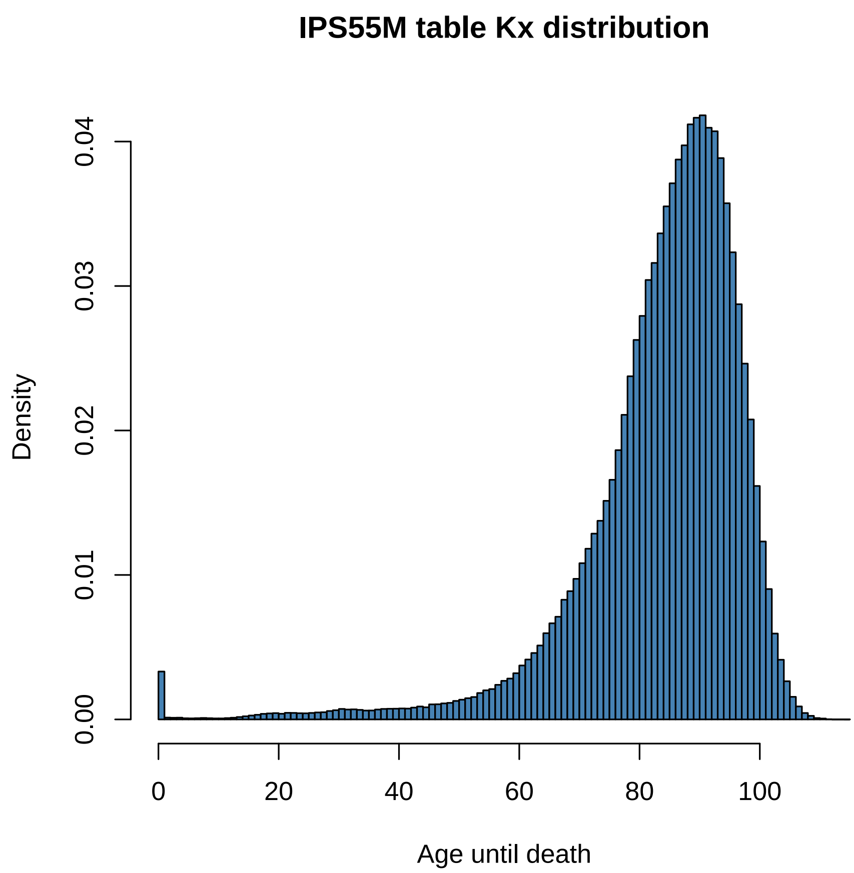

Figure 4: IPS55 deaths distribution function

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 $\tilde{Z}$ variates generation from term life insurances, increasing life term insurances, temporary annuity,and endowment insurances respectively. For each simulation, the unbiaseness is verified by comparing the mean of simulated variates with the theoretical APV. All simulations 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> numSim=50000
R> #term life insurance
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> t.test(x=sampleAxn,mu=APVAxn)$p.value

[1] 0.8255769

R> #increasing life insurance
R>
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> t.test(x=sampleIAxn,mu=APVIAxn)$p.value

[1] 0.9092557

R> #temporary annuity due
R>
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> t.test(x=sampleaxn,mu=APVaxn)$p.value

[1] 0.5544886

R> #endowment insurance
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> t.test(x=sampleAExn,mu=APVAExn)$p.value

[1] 0.941521
```

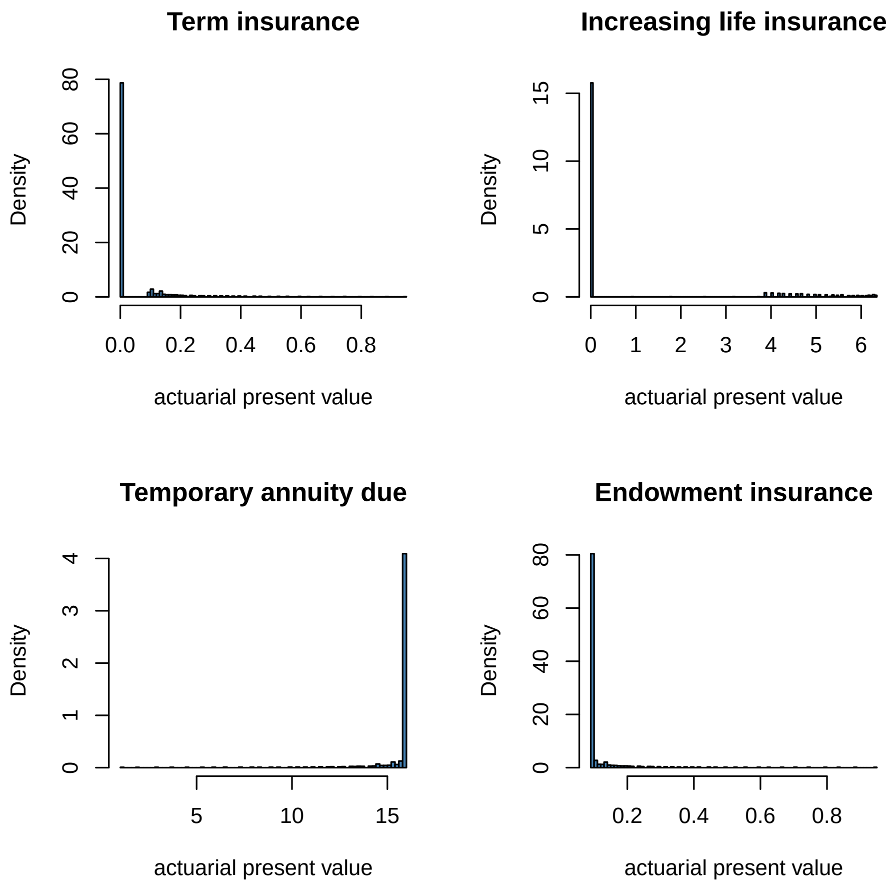

Figure 5: Life insurance stochastic variables distributions

The final example shows how the stochastic functions bundled in **lifecontingencies** can be used to make an actuarial appraisal of embedded benefits.

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, i.e. 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, Grndl, Schmidl, and Dorfman (2007), define the fair value of an insurance liability as the sum of its best estimate plus its risk margin. We have used parallel computation facilities bundled made available by package **parallel** due the computationally intensive calculation. Code has been adapted from examples of McCallum and Weston (2011) book.

```
R> #set the various parameters
R> employees=500
R> salaryDistribution=rlnorm(n=employees,m=10.77668944,s=0.086177696) #log-normal distribu
R> ageDistribution=round(runif(n=employees,min=25, max=65))
R> policyLength=sapply(65-ageDistribution, min,1)
R> #function to obtain the type of benefit
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> #configure the parallel library
R> #environment
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"))
R> #determine the best estimate of employees benefit
R> employeeBenefits=numeric(employees)
R> employeeBenefits<- parSapply(cl, 1:employees,getEmployeeBenefit, type="EV")
R> employeeBenefit=sum(employeeBenefits)
R> #determine the risk margin
R> nsim=100 #use 100 simulations
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> #summarize results
R> riskMargin=as.numeric(quantile(benefitDistribution,.75)-employeeBenefit)
R> totalBookedCost=employeeBenefit+riskMargin
R> employeeBenefit

[1] 141205.6

R> riskMargin

[1] 41337.51

R> totalBookedCost

[1] 182543.1
```

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 variates from both life tables and life insurances stochastic distributions.

One of the most important limitations of **lifecontingencies** is that it handles only single decrement tables. Another limitation is that currently it does not allow continuous time life contingencies to be modelled.
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 checkings with numerical examples reported in Bowers *et al.* (1997) 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 *et al.* (1997) 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 *et al.* (1997) 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.

Bowers N, Gerber H, Hickman J, Jones D, Nesbitt C (1997). "Actuarial Mathematics. Schaumburg." *IL: Society of Actuaries*.

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.

Dutang C, Goulet V, Pigeon M (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.

Mazzoleni P (2000). *Appunti di matematica attuariale*. EDUCatt Università Cattolica. ISBN 9788883110825. URL http://books.google.it/books?id=eGzSAAAACAAJ.

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

Post T, Grndl H, Schmidl L, Dorfman MS (2007). "Implications of IFRS for the European Insurance IndustryInsights 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.

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,(with discussion)." *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.

Ruckman C, Francis J (2006). *FINANCIAL MATHEMATICS: A Practical Guide for Actuaries and other Business Professionals*. Warren Centre for Actuarial Studies and Research, second edition edition.

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