Introduction to lifecontingencies Package

Giorgio Alfredo Spedicato
Universitá Cattolica del Sacro Cuore

Abstract

lifecontingencies performs actuarial present value calculation for life insurances. This paper briefly recapitulate the theory regarding life contingencies (life tables, financial mathematics and related probabilities) on life contingencies. Then it shows how **lifecontingencies** functions represent a perfect cookbook to perform life insurance actuarial analysis and related stochastic simulations.

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

1. Introduction

As of December 2011, **lifecontingencies** seems the first R package that deals with life insurance evaluation.
R has provided many package that actuaries can use within their professional activity.
However most packages are of mainly interest of non-life actuaries, due to the wider importante of regression modelling and distribution fitting in non - life than in life insurance. The package **actuar**, Dutang, Goulet, and Pigeon (2008), provides functions to fit loss distributions and to perform credibility analysis. Package **actuar** represents the computational side of the classical book Klugman, Panjer, Willmot, and Venter (2009). The package **ChainLadder**, Gesmann and Zhang (2011), provides functions to estimate non-life loss reserve. GLM analysis widely used in predictive modelling can be performed by the **base** package bundled within R. Specific models can be using the **gamlss** package, Rigby and Stasinopoulos (2005), or the **cplm** package, Zhang (2011).
Life actuaries conversely work more with demographic and financial data. R has a dedicated view to packages dedicated to financial analysis. However few packages exist to perform demographic analysis (as **demography**, Rob J Hyndman, Heather Booth, Leonie Tickle, and John Maindonald (2011), and **LifeTables**, Riffe (2011)). Finally no package exists that performs life contingencies calculations, as of December 2011.

Numerous commercial software specifically tailored to actuarial analysis are available in commerce. Moses and Prophet are currently the leading actuarial software in life insurance. This package aims to represent the R computational support of the concepts developed in the classical life contingencies book Bowers, Gerber, Hickman, Jones, and Nesbitt (1997). Since life contingencies theory grounds on demography and classical financial mathematics, I have made use of the Ruckman and Francis Ruckman and Francis and Broverman Broverman (2008) as references. The structure of the vignette document is:

1. Section 2 describes the underlying statistical and financial concepts regarding life contingencies theory.
2. Section 3 overviews the general structure of the package.
3. Section 4 gives a wide choice of **lifecontingencies** packages example.
4. Finally section 5 will provide a discussion of results and further potential developments.

2. Life contingencies statistical and financial foundations

Life insurance analysis involves the calculation of expected values of future cash flows, whose probabilities depend by events related to insured life contingencies and time value of money. Therefore life insurance actuarial mathematics grounds itself on concepts derived from demography and theory of interest (like present value).

A life table (also called a mortality table or actuarial table) is a table that shows, for each age x, the number of subjects l_x that are expected to be in life (at risk) at the beginning of age x. It represents a sequence of $l_0, l_1, \ldots, l_\omega$ being ω the farthest age that a person can obtain for the life table cohort. A life table is referred to a cohort of subjects, typically varying by sex, year of birth and country.

Many statistics can be derived from the l_x sequence. 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 $x+t$.
- ${}_tL_x = \sum\limits_{t=0}^{n} l_{x+t}$, 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 expected remaining lifetime for someone living at age x.

An exhaustive coverage of life table demographics can be found in Keyfitz and Caswell (2005). Life table are usually produced by institutions that have access to large amount of reliable historical data, like official statistics or social security bureaus. Actuaries often start from those table and modify underlying survival probabilities to make the table better fit to the insured pool experience. Life table analysis provides the empirical data to assess the time - until - death random variables for any individual aged x, being T_x and K_x the continuous and curtate form respectively.

Classical financial mathematics deals with monetary amount that could be available in different times. The most important concept in classical financial mathematics is the present

value (see formula 2). 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 rates, i_t, represents the measure of price of money per unit of time. Formula 1 shows the relationship between the accumulation function, $A(t)$ and interest and discount rates, both effective and nominal.

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

All financial mathematics functions (as annuities, $\bar{a}_{\overline{n}|}$, or accumulated values, $s_{\overline{n}|}$) can be written as a particular case of formula 2.

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

Actuaries uses 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, but the time and the final values depend by events regarding the life of the insured head. Their expected value is named actuarial present value (APV). While APV is certainly the most important statistic used by actuaries, as long as it represents the average cost of the coverage the insurer provides, **lifecontingencies** provides functions to assess the distribution of life contingencies quantities as random value generation.

Some examples of life contingencies follow. The term life insurances represent a contract where a sum b_t is payable whether the insured head dies within n years. A flat n - term life insurance is expressed in formula 3 and its APV symbol is $A^1_{x:\overline{n}|}$.

$$Z_n = \begin{cases} b_{\tilde{T}} v^{\tilde{T}} \tilde{T} \leq n \\ 0 \tilde{T} > n \end{cases} \tag{3}$$

Another example is the annuity due, $\ddot{a}_x$, that consists in a series of cash flows of equal amounts payable at the beginning of each period until death. Formula 4 expresses its APV.

$$\sum_{k=0}^{\infty} a_{\overline{k+1}|} \, {}_k p_x q_{x+k} \tag{4}$$

The pure endowment is a stochastic variable defined in formula 5. Its APV symbol is ${}_nE_x$. Under a pure endowment contract a sum is paid after n year if the insured aged x still lives at age $x+n$.

$$\begin{cases} v^T, \tilde{T} > n \\ 0, \tilde{T} \leq n \end{cases} \tag{5}$$

The **lifecontingencies** package provides functions that allows the user to evaluate standard life insurance contract APV. Function for A_x (life insurance), ${}_nE_x$ (the pure endowment), $\ddot{a}_x$ (the annuity due), $(DA)^1_{x:\overline{n}|}$ (the decreasing term life insurance) and $(IA)_x$ (increasing term life insurance) are available as long as variants. Most important variants consist in allowing fractional terms and temporary duration.

.

In general present values can be expressed as the scalar product of full value cash flows $\bar{c}$ and their corresponding discount rate $\bar{v}$. APV formula taxes into account the present value. While most financial mathematics and lifecontingencies formulas can be expressed in symbolic form, **lifecontingencies** will evaluate present values and APVs using vector calculus. Therefore **lifecontingencies** builds the vectors of payments, $\bar{c}$, discounts, $\bar{v}$ and probabilities, $\bar{p}$ and it evaluates their scalar product as shown by equation 6.

$$\langle\langle\bar{c}\bullet\bar{v}\rangle\bullet\bar{p}\rangle \tag{6}$$

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" [package "lifecontingencies"]

Slots:

Name:          x        lx      name
Class:   numeric   numeric character

Known Subclasses: "actuarialtable"
```

Class `actuarialtable` inherits from `lifetable` class and has another additional slots,the interest rate.

```
R> showClass("actuarialtable")

Class "actuarialtable" [package "lifecontingencies"]

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, financial mathematics and life contingencies analysis functions.

The demographic group comprises the following functions:

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 following functions, for which we report most important function:

1. `presentValue` returns the present value for a series of cash flows.
2. `annuity` returns the present value of a annuity - certain, $a_{\overline{n}|}$.
3. `iecreasingAnnuity` returns the present value of an increasing annuity - certain, $(IA)_n$.
4. `accumulatedValue` returns the future value of a series of cash flows, $s_{\overline{n}|}$.
5. `decreasingAnnuity` returns the present value of an increasing annuity, $(DA)_{\overline{n}|}$.
6. `accumulatedValue` returns the future value of a payments sequence, $s_{\overline{n}|}$.
7. `nominal2Real` returns the effective annual interest (discount) rate i given the nominal m-periodal interest $i^{(}m)$ or discount d^m rate.
8. `real2Nominal` returns the m-periodal interest or discount rate given the m periods or the discount.
9. `intensity2Interest` returns the intensity of interest δ given the interest rate i.
10. `interest2Intensity` returns the interest rate i given the intensity of interest δ.

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

1. `Axn` returns the APV for life insurances.
2. `Axyn` returns the APV for two heads life insurances.
3. `axn` returns the APV for annuities.
4. `axyn` returns the APV for two heads annuities.

5. `Exn` returns the APV for the pure endowment.

6. `Iaxn` returns the APV fof the increasing annuity.

7. `IAxn` returns the APV fof the increasing life insurance.

8. `DAxn` returns the APV for the decreasing life insurance.

4. Code and examples

4.1. Classical financial mathematics example

The **lifecontingencies** package provides functions to perform classical financial analysis. Following examples will show how to handle interest and discount rates with different compounding frequency, how to perform present value, annuities and future values analysis calculations, loans amortization and bond pricing.

Interest rate functions

Following examples show how to switch from $i^m \to i$

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

[1] 0.02959524

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

[1] 0.06167781

R> #APR
R> #4% per year compounded quarterly is
R> nominal2Real(0.04,4)

[1] 0.04060401

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

[1] 3.941363

R> #nominal interest rate (in 100s) compounded quarterly
```

and from $d^m \to d$

```
R> #a nominal rate of discount of 4% payable quarterly is equal to a
R> real2Nominal(i=0.04,m=12,type="discount")

[1] 0.04075264
```

Present value analysis

Performing a project appraisal means evaluating the present value of all net cash flows, as shown in code below:

```
R> #suppose an investment requires and grants following cash flows
R> capitals=c(-1000,200,500,700)
R> #at time (vector) t.
R> times=c(-2,-1,4,7)
R> #the preset value of the investment is
R> presentValue(cashFlows=capitals, timeIds=times,
+              interestRates=0.03)

[1] 158.5076

R> #assuming 3% interest rate
R>
R>   #while if interest rates were time - varying
R> #e.g. 0.04 0.02 0.03 0.057
R> presentValue(cashFlows=capitals, timeIds=times,
+              interestRates=c( 0.04, 0.02, 0.03, 0.057))

[1] 41.51177

R> #and if the last cash flow is uncertain, as we assume a
R> #receiving probability of 50%
R> presentValue(cashFlows=capitals, timeIds=times,
+  interestRates=c( 0.04, 0.02, 0.03, 0.057),
+              probabilities=c(1,1,1,0.5))

[1] -195.9224
```

Annuities and future values

Example of $a_{\overline{n|}}$ and $s_{\overline{n|}}$ evaluations are reported below.

```
R> #PV annuity immediate 100$ each year 5 years @9%
R> 100*annuity(i=0.09,n=5)

[1] 388.9651

R> #while the corresponding future values is
R> 100*accumulatedValue(i=0.09,n=5)

[1] 598.4711

R> #A man wants to save 100,000 to pay for the education
R> #of his son in 10 years time. An education fund requires the investors to
R> #deposit equal instalments annually at the end of each year. If interest of
R> #0.075 is paid, how much does the man need to save each year (R) in order to
R> #meet his target?
R> 100000/accumulatedValue(i=0.075,n=10)
```

```
[1] 7068.593
```

while the code below shows how fractional annuities ($a^{(m)}_{lcroofn}$) can be handled within `annuity` and `accumulatedValue` functions.

```
R> #Find the present value of an annuity-immediate of
R> #100 per quarter for 4 years, if interest is compounded semiannually at
R> #the nominal rate of 6%.
R> #the APR is
R> APR=nominal2Real(0.06,2)
R> 100*4*annuity(i=APR,n=4,m=4)

[1] 1414.39
```

Finally `increasingAnnuity` and `decreasingAnnuity` functions handle increasing ($(IA)_x$) and decreasing ($(DA)_x$) annuities.

```
R> #An increasing n-payment annuity-due shows payments of 1, 2,
R> #... , n
R> #at time 0, 1, ... ,
R> #n - 1 . At interest rate of
R> #0.03 and n=10, its present value of the annuity is
R> increasingAnnuity(i=0.03, n=10,type="due")

[1] 46.18416

R> #while the present value of a decreasing
R> #annuity due of 10, 9,...,1
R> #from time 1 to time 10 is
R> decreasingAnnuity(i=0.03, n=10,type="immediate")

[1] 48.99324
```

Finally the calculation of the present value of a geometrically increasing annuity is shown in the code below

```
R> #assume each year the annuity increases its value by 3%
R> #while the interest rate is 4%
R> #first determine the effective interest rate
R> ieff=(1+0.04)/(1+0.03)-1
R> #assume the annuity lasts 10 years
R> annuity(i=ieff,n=10)

[1] 9.48612
```

Loan amortization

The code lines below show how an investment amortization schedule will be repaired.

Suppose loaned capital is C, then assuming an interest rate i, the amount due to the lender at each instalment is $R = \frac{C}{a_{\overline{n}|}}$.

At each installment the R_t installment repays $I_t = C_{t-1} * i$ as interest and $C_t = R_t - I_t$ as capital.

```
R> capital=100000
R> interest=0.05 #assume 5% effective annual interest
R> payments_per_year=2 #payments per year
R> rate_per_period=(1+interest)^(1/payments_per_year)-1
R> years=5 #five years length of the loan
R> installment=1/payments_per_year*capital/annuity(i=interest, n=years,m=payments_per_year
R> installment

[1] 11407.88

R> #compute the balance due at the begin of 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)-installment
+          cat("Payment ",i, " balance due:",round(balance_due[i]),"\n")
+  }

Payment  2  balance due: 81903
Payment  3  balance due: 72517
Payment  4  balance due: 62900
Payment  5  balance due: 53046
Payment  6  balance due: 42948
Payment  7  balance due: 32600
Payment  8  balance due: 21998
Payment  9  balance due: 11133
Payment  10  balance due: 0
```

Bond pricing

Bond pricing is another application of present value analysis. A standard bond whose principal will be repaid at time T is a series of coupon c_t, priced according to a coupon rate $j^(k)$ on a principal C. Formula 7 expresses the present value of a bond.

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

We will show how to evaluate a standard bond with following examples:

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

[1] 1029.25

R> #bond coupon rate 3%, one coupons per year, face value 1000, yield 3%, three years to m
R> bond(1000,0.06,1,0.06,3)

[1] 1000
```

4.2. Lifetables and actuarial tables analysis

`lifetable` classes represent the basic class designed to handle life table calculations. A `actuarialtable` class inherits from `lifetable` class adding one more slot to set the a priori rate of interest.
Both classes have been designed using the S4 class framework.
Examples follow showing how `lifetable` and `actuarialtable` objects initialization, basic survival probability and life tables 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 items are needed:

1. The years sequence, that is an integer sequence $0, 1, \ldots, \omega$. It shall starts from zero and going to the 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.
3. The name of the life table.

```
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> fakeLt=new("lifetable",x=x_example, lx=lx_example, name="fake lifetable")
```

A `print` (or `show`) method are available. These methods report the x, lx, px and ex in tabular form.

```
R> print(fakeLt)

Life table fake 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 adapted to `lifetable` classes, as code below shows.

```
R> #show head method
R> head(fakeLt)
```

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

```
R> #show tail method
R> tail(fakeLt)
```

```
   x  lx
5  4 680
6  5 600
7  6 550
8  7 400
9  8 200
10 9  50
```

Nevertheless the easiest way to create a `lifetable` object is starting from a suitable existing `data.frame`.

```
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 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>          #create the 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, name="IPS 55 Males")
R>          ips55F=new("lifetable",x=xIPS55F, lx=lxIPS55F, name="IPS 55 Females")
```

The last way a lifetable object can be created is generating it from one year survival or death probabilities. Such probabilities could be obtained from mortality projection methods (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,type="qx",
```

An actuarialtable class inherits from the lifecontingencies class, but it contains and additional slot: the interest rate slot. slot.

```
R>          #assume 3% interest rate
R>          fakeAct=new("actuarialtable",x=fakeLt@x, lx=fakeLt@lx, interest=0.03,
+                          name="fake actuarialtable")
```

Method getOmega provides the ω age.

```
R>          getOmega(fakeAct)
```

```
[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(fakeLt)
```

```
Life table fake 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(fakeAct)
```

```
Actuarial table  fake 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
```

Basic demographic calculations

Basic probability calculations may be performed on valid `lifetable` or `actuariatable` objects. Below calculations for ${}_tp_x$, ${}_tq_x$ and $\mathring{e}_{x:\overline{n}|}$.

```
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 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, constant force of mortality and hyperbolic assumption.

```
R> data(soa08Act) #load Society of Actuaries illustrative life table
R> pxt(soa08Act,80,0.5,"linear") #linear interpolation (default)

[1] 0.9598496

R> pxt(soa08Act,80,0.5,"constant force") #constant force

[1] 0.9590094

R> pxt(soa08Act,80,0.5,"hyperbolic") #hyperbolic Balducci's assumption.

[1] 0.9581701
```

Analysis of two heads survival probabilities can be performed also, as shown by code below:

```
R> pxyt(fakeLt,fakeLt,x=6, y=7, t=2) #joint survival probability

[1] 0.04545455

R> pxyt(fakeLt,fakeLt,x=6, y=7, t=2,status="last") #last survival probability

[1] 0.4431818

R> #evaluate the expected joint life time for a couple aged 65 and 63 using Italina IPS55
R> exyt(ips55M, ips55F, x=65,y=63, status="joint")

[1] 19.1983
```

4.3. Classical actuarial mathematics examples

Classical actuarial mathematics on life contingencies will follow now. We will use the SOA illustrative life table on all following examples.

Life insurance examples

Following examples show the APV (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.
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> #The APV of a life insurance for a 10-year term life insurance for an
R> #insured aged 40 @ 4% interest rate is
R> Axn(soa08Act, 30,10,i=0.04)

[1] 0.01577283

R> #same as above but payable at the end of month of death
R> Axn(soa08Act, x=30,n=10,i=0.04,k=12)

[1] 0.01605995

R> #a whole life for a 40 years old insured at @4% is
R> Axn(soa08Act, 40) #soa08Act has 6% implicit interest rate

[1] 0.1613242

R> #a 5-year deferred life insurance, 10 years length, 40 years age, @5% interest rate
R> Axn(soa08Act, x=40,n=10,m=5,i=0.05)

[1] 0.03298309

R> #Five years annually decreasing term life insurance, age 50.
R> DAxn(soa08Act, 50,5)

[1] 0.08575918

R> #Increasing 20 years term life insurance, age 40
R> IAxn(soa08Act, 40,10)

[1] 0.1551456
```

while following code evaluates pure endowments APV, ${}_{n}E_{x}$, assuming SOA life table at 6% interest rate.

```
R> #evaluate the APV for a n year pure endowment, age x=30, n=35, i=6%
R> Exn(soa08Act, x=30, n=35, i=0.06)

[1] 0.1031648

R> #try i=3%
R> Exn(soa08Act, x=30, n=35, i=0.03)

[1] 0.2817954
```

Life annuities examples

Following examples show annuities APV calculations for

1. annuity immediate for a subject aged 65, a_{65}.
2. annuity due for a subject aged 65, $\ddot{a}_{65}$.
3. 20 years annuity due with monthly fractional payments of \$1000, $\ddot{a}^{(12)}_{65:\overline{20}|}$.

All examples assume SOA life table at 6% interest rate.

```
R> #assuming insured's age x=65 and SOA illustrative life table @6% hold for all examples
R> #annuity immediate
R> axn(soa08Act, x=65, m=1)

[1] 8.896928

R> #annuity due
R> axn(soa08Act, x=65)

[1] 9.896928

R> #due with monthly payments of $1000 provision
R> 12*1000*axn(soa08Act, x=65,k=12)

[1] 113179.1

R> #due with montly payments of $1000 provision, 20 - years term
R> 12*1000*axn(soa08Act, x=65,k=12, n=20)

[1] 108223.5

R> #immediate with monthly payments of 1000 provision, 20 - years term
R> 12*1000*axn(soa08Act, x=65,k=12,n=20,m=1/12)

[1] 107321.1
```

Benefit premiums examples

lifecontingencies package functions can be used to evaluate benefit premium for life contingencies, using the formula ${}_hP^1_{x:\overline{n}|} = APV \ddot{a}_{x:\overline{h}|}$.

```
R> data(soa08Act) #use SOA MLC exam illustrative life table
R> #Assume X, aged 30, whishes to buy a 250K 35-years life insurance
R> #premium paid annually for 15 years @2.5%.
R> Pa=100000*Axn(soa08Act, x=30,n=35,i=0.025)/axn(soa08Act, x=30,n=15,i=0.025)
R> Pa

[1] 921.5262

R> #if premium is paid montly
R> Pm=100000*Axn(soa08Act, x=30,n=35,i=0.025)/axn(soa08Act, x=30,n=15,i=0.025,k=12)
R> Pm

[1] 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)
```

Benefit reserves examples

Now we will evaluate the benefit reserve for a 20 year 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 is P, determined from equation

$$P\ddot{a}_{40:\overline{20}|} = 100000 A^{\,1}_{40:\overline{20}|}$$

. The benefit reserve is ${}_kV^{\,1}_{40+t:\overline{n-t}|} = 100000 A^{\,1}_{40+t:\overline{20-t}|} - P\ddot{a}_{40+t:\overline{20-t}|}$ for $t = 0 \ldots 19$.

```
R>          P=100000*Axn(soa08Act,x=40,n=20,i=0.03)/axn(soa08Act,x=40,n=20,i=0.03)
R>          for(t in 0:19) cat("At time ",t," benefit reserve is ", 100000*Axn(soa08Act,x=4

At time  0  benefit reserve is  0
At time  1  benefit reserve is  306.9663
At time  2  benefit reserve is  604.0289
At time  3  benefit reserve is  889.0652
At time  4  benefit reserve is  1159.693
At time  5  benefit reserve is  1413.253
At time  6  benefit reserve is  1646.808
At time  7  benefit reserve is  1857.044
```

```
At time  8  benefit reserve is  2040.286
At time  9  benefit reserve is  2192.436
At time  10  benefit reserve is  2308.88
At time  11  benefit reserve is  2384.513
At time  12  benefit reserve is  2413.576
At time  13  benefit reserve is  2389.633
At time  14  benefit reserve is  2305.464
At time  15  benefit reserve is  2152.963
At time  16  benefit reserve is  1922.973
At time  17  benefit reserve is  1605.162
At time  18  benefit reserve is  1187.872
At time  19  benefit reserve is  657.8482
```

The benefit reserve for a whole life annuity with level annual premium is ${}_kV({}_{n|}\ddot{a}_x)$, that equals ${}_{n|}\ddot{a}_x - \bar{P}({}_{n|}\bar{a}_x)\ddot{a}_{x+k:\overline{n-k|}}$ when $x \dots n$, $\ddot{a}_{x+k}$ otherwise. The figure is shown in 1.

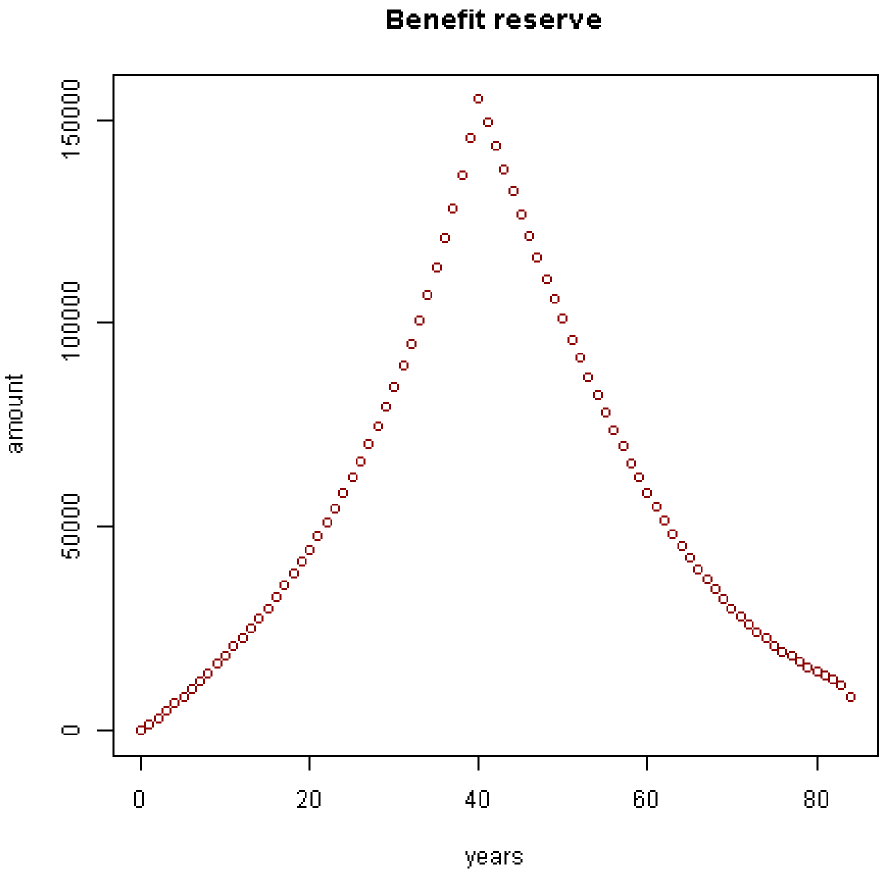

Figure 1: Benefit reserve of $\ddot{a}_{65}$

Insurance and annuities on two heads

Lifecontingencies package provides functions to evaluate life insurance and annuities on two lifes. Following examples will 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)-axyn(soa08Act,soa08Act, x=65,y=
```

```
[1] 10.35704
```

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

```
[1] 10.35704
```

Reversionary annuity (annuities payable to life y upon death of x), $a_{x|y} = a_y - a_{xy}$ can also be evaluate using **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 paragraphs will show some stochastic analysis that can be performed by our package, both in demographic analysis and life insurance evaluation.

The age-until-death, both in the continuous (T_x) or curtate form (K_x), is a stochastic variable whose distribution is implicit within the deaths distribution of a given 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.

```
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"),mu=exn(soa08Act, x=29,type="curta

[1] 0.6378775

R>          #statistically not significant
```

Finally figure 2 shows the deaths distribution implicit in the ips55M life table.

The APV is a present value of a random variable that represents a composite function between the discount amount and indicator variables regarding the life status of the insured. Figure 3 shows the stochastic distribution of $\ddot{a}_{65}$.

5. Discussion

The **lifecontingencies** package allows actuaries to perform financial and life contingencies actuarial mathematics within R. It offers the basic tools to manipulate life tables and perform financial calculations. Pricing, reserving and stochastic evaluation of most important life insurance contract can be performed within R.

Future work spans in multiple directions. Currently **lifecontingencies** handles only one - year life tables and single causes of decrement only. We expect to make the package more flesible

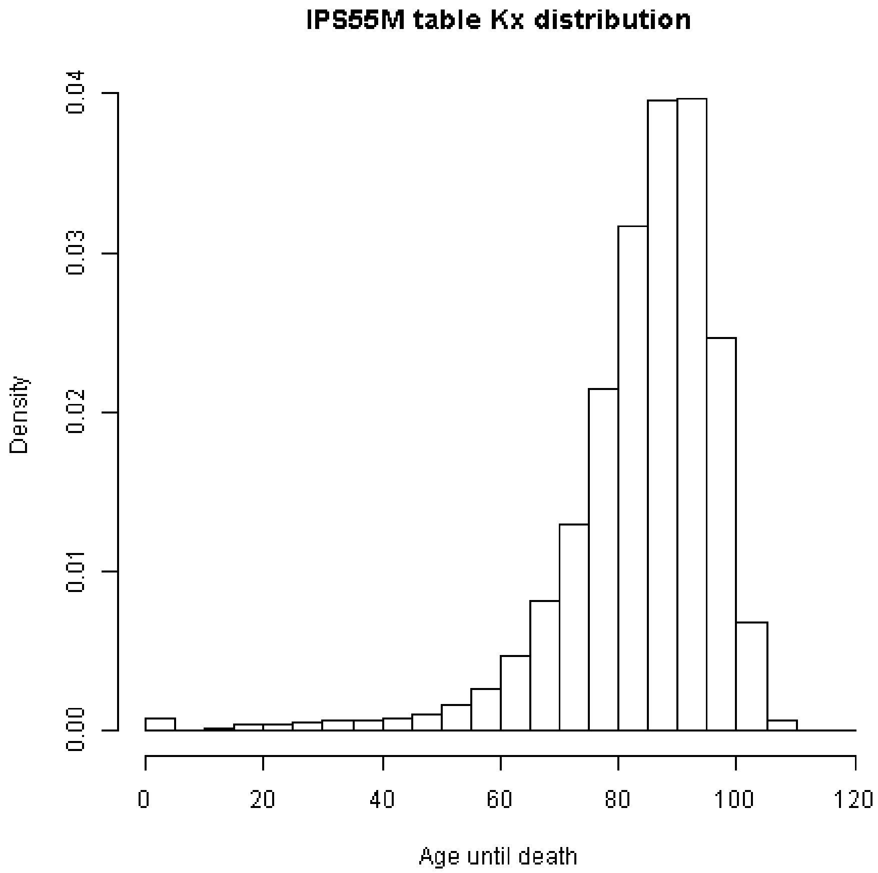

Figure 2: Deaths distribution implicit in the IPS55 males table

from this side. The stochastic calculation modules will be improved, morover. Similarly we expect to integrate C++ fragments using **Rcpp** package whether it effectively improves performance.
Finally we wish to provide **lifecontingencies** coerce methods toward packages specialized in demographic analysis, like **demography** and **LifeTables** and interest rates modelling.

Disclaimer

The accuracy of calculation have been verified by checkings with numerical examples reported in Bowers *et al.* (1997). The package numerical results are identical to those reported in the Bowers *et al.* (1997) 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) uses an analytical formula.

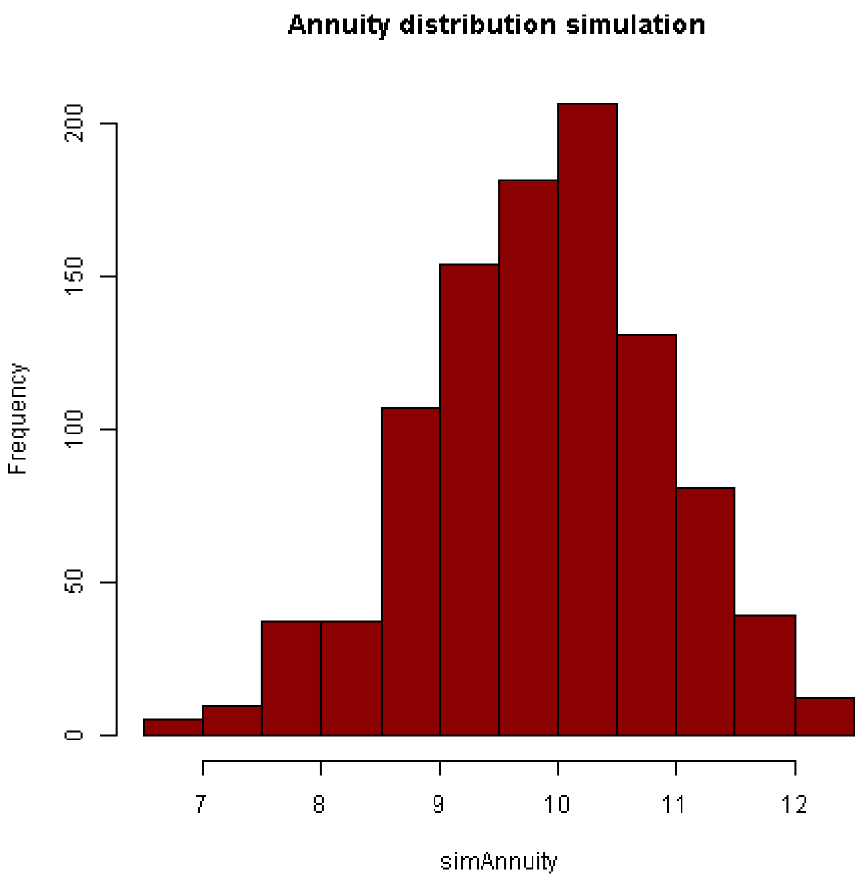

Figure 3: Stochastic distribution of $\ddot{a}_{65}$

Acknowledgments

I wish to thank Christophe Dutang and Tim Riffle for their valuable suggestions.

References

Bowers N, Gerber H, Hickman J, Jones D, Nesbitt C (1997). "Actuarial Mathematics. Schaumburg." *IL: Society of Actuaries*, pp. 79–82.

Broverman S (2008). *Mathematics of investment and credit*. ACTEX academic series. ACTEX Publications. ISBN 9781566986571. URL http://books.google.it/books?id=lK5WDvdc7TcC.

Chambers J (2008). *Software for data analysis: programming with R*. Statistics and computing. Springer. ISBN 9780387759357. URL http://books.google.com/books?id=UXneuOIvhEAC.

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.

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

Keyfitz N, Caswell H (2005). *Applied mathematical demography*. Statistics for biology and health. Springer. ISBN 9780387225371. URL http://books.google.it/books?id=PxSVxES7SjOC.

Klugman S, Panjer H, Willmot G, Venter G (2009). *Loss models: from data to decisions*. Third edition. Wiley New York.

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 (????). "FINANCIAL MATHEMATICS:A Practical Guide for Actuaries and other Business Professionals."

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