Introduction to lifecontingencies Package

Giorgio Alfredo Spedicato
StatisticalAdvisor Inc

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 September 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 actuarial side, where statistics take a wider share of the day-to-day work. The package **actuar**, Dutang, Goulet, and Pigeon (2008), provides functions to fit loss distributions and to perform credibility analysis. It 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 even if interesting applications can be build by **gamlss**, Rigby and Stasinopoulos (2005), or by the package **cplm**, Zhang (2011).
On the other hand, life actuaries works 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 (see for examples **demography**, **?**, and **LifeTables**, **?**) as of September 2011 no package exists to perform life contingencies calculation.
Numerous commercial packages are available to conduct actuarial analysis both in life and non - life site. Currently Tower Watson firm produces the most used actuarial packages. 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).

The structure of the vignette document is:

1. Section 2 describes the underlying statistical and financial concepts regarding the life contingencies.

2. Section 4 gives a wide choice of lifecontingencies packages example.

3. Finally section 5 will provide a discussion of results and further potential developments.

2. The statistics of life contingencies actuarial evaluation

Life insurance analysis involves the calculation of expected values of future cash flows, whose probabilities depend by events related to insureds life contingencies. Therefore life insurance actuarial mathematics uses concepts derived from demography (as life table probability calculations) and theory of interest (like present value).

A life table (also called a mortality table or actuarial table) consists is a table which shows, for each age x, the number of subjects l_x of that analyzed coort that are expected to be in life at the beginning of that age. Therefore it represents a sequence of $l_0, l_1, \ldots, l_\omega$ being ω the fartest age that a person can obtain.
Many quantities can be derived from the l_x sequence. A non exaustive 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 $x+t$.
- ${}_tL_x = \sum_{t=0}^{n} l_{x+t}$, the expected number of years lived by the coohort 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 somone living at age x.

An exaustive 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 bureau or social security. Actuaries often start from those table and modify underlying survival probabilities to make the table better fit to the insureds pool experience.

Financial mathematic deals with monetary amount that could be available in different times and whose possession is not certain. Probably the most important concept in classical financial mathematics is the present value (see formula 2), that represent the currently valued figure for a series of cash flows CF_t available in different periods of time using interest rates i_t as the measure of price of money per unit of time. Formula 1 shows the relationship between interest and discount rates, both effective and nominal.

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

All financial matemathic functions (as annuities, $\bar{a}_{\overline{n}|}$, or accumulated values, $s_{\overline{n}|}$) can be seen as an adapted version 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 evalutate the expected value of insured cash flows, obtaining quantities called Actuarial Present Values (APV). E.g. in term life insurance, $A^1_{x:\overline{n}|}$, the insured amount of is payable only if the insured death dies within age x and $x+t$. Another example is the annuity, $\ddot{a}_x$, that consists in a series of cash flows of equal amounts payable at the beginning of each periods until the insured dies. The **lifecontingencies** package contains 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 (fractional periods and differing terms). It is worth to remeber that life contingencies is a stochastic value: the life insurance is the random variable $v^{\tilde{T}_x}$ being $\tilde{T}$ the curtate remaining life time and v the uni periodal discount factor. **lifecontingencies** contains formulas to drawn random samples from life contingencies distributions.

3. The structure of the package

Package **lifecontingencies** contains classes and methods to handle lifetable in a way convenient for actuaries.
Moreover it bundles financial mathematics functions to help the analyst to perform present value analysis. Finally most used actuarial functions to evaluate lifecontingencies insurance, as reported in the classical book Bowers *et al.* (1997), have been made available.

The package is load 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"
```

Functions are available to evaluate actuarial present values for life contingencies functions as $\ddot{a}^{(m)}_{x:\overline{n}|}$, $A^{1}_{x:\overline{n}|}$, $A_{x:\overset{1}{\overline{n}|}}$, $(DA)^{1}_{x:\overline{n}|}$ and $(IA)^{1}_{x:\overline{n}|}$.
Some functions allows to return the simulated value of most life contingencies functions.
Demos and vignettes (like this document) are also available.

4. Code and examples

4.1. Classical financial mathematics example

The lifecontingencies package provides function to perform classical financial analysis. Functions `real2Nominal` and `nominal2Real` allows to switch easily from nominal to effective APR easily. Functions `annuity` and `accumulatedValue` calculates the values of $a_{\overline{n|}}$ and $s_{\overline{n|}}$.

Interest rate functions

```
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
```

Present value analysis

```
R>         #say we are at time t0, and following capitals would have been received (+) / p
R>         #at time (vector) t.
R>         capitals=c(-1000,200,500,700)
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>         #@3% interest rate
R>         #while if interest rates were time - varying 0.04 0.02 0.03 0.057
R>         presentValue(cashFlows=capitals, timeIds=times, interestRates=c( 0.04, 0.02, 0.

[1] 41.51177

R>         #and if the last cash flow is uncertain, as we assume a receiving probability o
R>         presentValue(cashFlows=capitals, timeIds=times, interestRates=c( 0.04, 0.02, 0.

[1] -195.9224
```

Loan amortization

```
R> capital=100000
R> interest=0.05 #assume 5% of interest
R> payments_per_year=2 #montly paymentsa
R> montlyRate=(1+interest)^(1/payments_per_year)-1
R> years=10 #ten years length of the loan
R> installment=capital/annuity(i=interest, n=years,m=payments_per_year)
R> installment

[1] 6396.251

R> #compute the balance fue
R> balance_due=numeric(years*payments_per_year)
R> balance_due[1]=capital*(1+montlyRate)-installment
R> for(i in 2:length(balance_due))
+  {
+          balance_due[i]=balance_due[i-1]*(1+montlyRate)-installment
+          cat("Payment ",i, " balance due:",round(balance_due[i]),"\n")
+  }

Payment  2  balance due: 92050
Payment  3  balance due: 87926
Payment  4  balance due: 83702
Payment  5  balance due: 79372
Payment  6  balance due: 74936
Payment  7  balance due: 70390
Payment  8  balance due: 65733
Payment  9  balance due: 60960
Payment  10  balance due: 56069
Payment  11  balance due: 51057
Payment  12  balance due: 45922
Payment  13  balance due: 40659
Payment  14  balance due: 35267
Payment  15  balance due: 29742
Payment  16  balance due: 24080
Payment  17  balance due: 18279
Payment  18  balance due: 12334
Payment  19  balance due: 6242
Payment  20  balance due: 0

R>
```

Saving account projection

```
R>          #assume the bank will grant a yearly interest of 2.5% on effectively
R>          #invested amounts
R>          #the bank will charge a service charge of $1 and a service fee of
R>          #0.01 on the amount between 0 and 100, 0.005 on the amounts between 100 and 150
R>
R>
R>          cumulatedSavings<-function(amount, rate, periods)
+           {
+                   service_charge=1
+                   service_fee=(0.01*min(100,amount)+0.005*max(0,min(50,amount-100)))
+                   invested_amount=amount-service_charge-service_fee
+                   out=invested_amount*accumulatedValue(i=rate, periods=periods)
+                   return(out)
+           }
R>          savings_sequence=seq(from=50, to=300, by=10) #possible montly savings
R>          periods=30*12 #suppose 30 years of savings
R>          yearly_rate=0.025 #suppose a APR of 2.5 that is a
R>          montly_effective_rate=(1+yearly_rate)^(1/12)-1
R>          cumulated_value=sapply(savings_sequence, cumulatedSavings, montly_effective_rat
R>          #plot(savings_sequence, cumulated_value, type="l")
```

4.2. Functions to switch between nominal and effective interest rates

```
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
```

4.3. Working with lifetable and actuarial table objects

Lifetable objects represent the basic class designed to handle life table calculations needed to evaluate life contingencies. Actuarialtable class inherits from lifetable class.
Both have been designed using the S4 class framework. 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 - equivalent) method is also available, reporting 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
```

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

```
R>          irate=0.03
R>          fakeAct=new("actuarialtable",x=fakeLt@x, lx=fakeLt@lx, interest=irate, name="fa
```

Currently just one method, `getOmega` has been implemented for lifetable and actuarialtable S4 classes, that provides the ω age.

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

```
[1] 9
```

Nevertheless the easiest way to create a `lifetable` object is to start from a `data.frame`.

```
R>          data(demoUsa) #load USA Social Security LT
R>          usaMale07=demoUsa[,c("age", "USSS2007M")]
R>          usaMale00=demoUsa[,c("age", "USSS2000M")]
R>          #coercing 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"
```

An other way to obtain lifetable object is to generate them from one year survival or death probabilities. These probabilities could for example 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", nam
```

4.4. Survival distribution and life tables

After a lifecontingencies table has been created, basic probability calculations may be performed. Below calculations for ${}_tp_x$, ${}_tq_x$ and $\mathring{e}_{x:\overline{n}|}$.

```
R>          pxt(fakeLt,2,1) #probability to survive one year, being at age 2
```

```
[1] 0.8235294
```

```
R>          qxt(fakeLt,3,2) #probability to die within two years, being at age 3
```

```
[1] 0.1428571
```

```
R>          exn(fakeLt, 5,2) #expected life time between 5 an 7 years
```

```
[1] 1.583333
```

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") #constant force

[1] 0.9581701
```

Analysis of two heads survival probabilities are possible:

```
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
```

If we want a more real example, lets use the IPS55 Italian population life table

```
R> #create the tables
R>
R>         lxIPS55M<-with(demoIta, IPS55M)
R>         pos2Remove<-which(lxIPS55M %in% c(0,NA))
R>         lxIPS55M<-lxIPS55M[-pos2Remove]
R>         xIPS55M<-seq(0,length(lxIPS55M)-1,1)
R>         lxIPS55F<-with(demoIta, IPS55F)
R>         pos2Remove<-which(lxIPS55F %in% c(0,NA))
R>         lxIPS55F<-lxIPS55F[-pos2Remove]
R>         xIPS55F<-seq(0,length(lxIPS55F)-1,1)
R>         ips55M=new("lifetable",x=xIPS55M, lx=lxIPS55M, name="IPS 55 Males")
R>         ips55F=new("lifetable",x=xIPS55F, lx=lxIPS55F, name="IPS 55 Females")
R>         #implicit omega age
R>
R>         getOmega(ips55M) #for males

[1] 117
```

```
R>          getOmega(ips55F) #for females

[1] 118

R>          #evaluate the joint expected life time for a couple
R>          #male ages 65 and females ages 63
R>          exyt(ips55M, ips55F, x=65,y=63, status="joint")

[1] 19.1983

R>
```

4.5. Classical actuarial mathematics examples

We will now show some classical actuarial mathematics example regarding the evaluation of actuarial present value (APV) of some life insurance benefits, benefit premiums and benefit reserves for classical life insurances.
For all reported examples, we will use the SOA illustrative life table and the insured amount is considered equal to 1 unless otherwise specified.

Life insurance examples

Following examples show APV for a series of life insurances.

```
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(actuarialtable=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 examples evaluate pure endowments

```
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 (immediate, due, with fractional payments provision, deferred, etd ...) APV calculations.

```
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 montly 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 montly 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

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

. The benefit reserve is ${}_kV^{\ 1}_{40+t:\overline{n-t}|} = 100000A^{\ 1}_{40+t:\overline{20-t}|} - P\ddot{a}_{40+t:\overline{20-t}|}$ 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 $x \ldots n$, $\ddot{a}_{x+k}$ otherwise. The figure is shown in 3.

Insurance and annuities on two heads

Lifecontingencies package provides function 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), $a_{x|y} = a_y - a_{xy}$ are also evaluable.

```
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
```

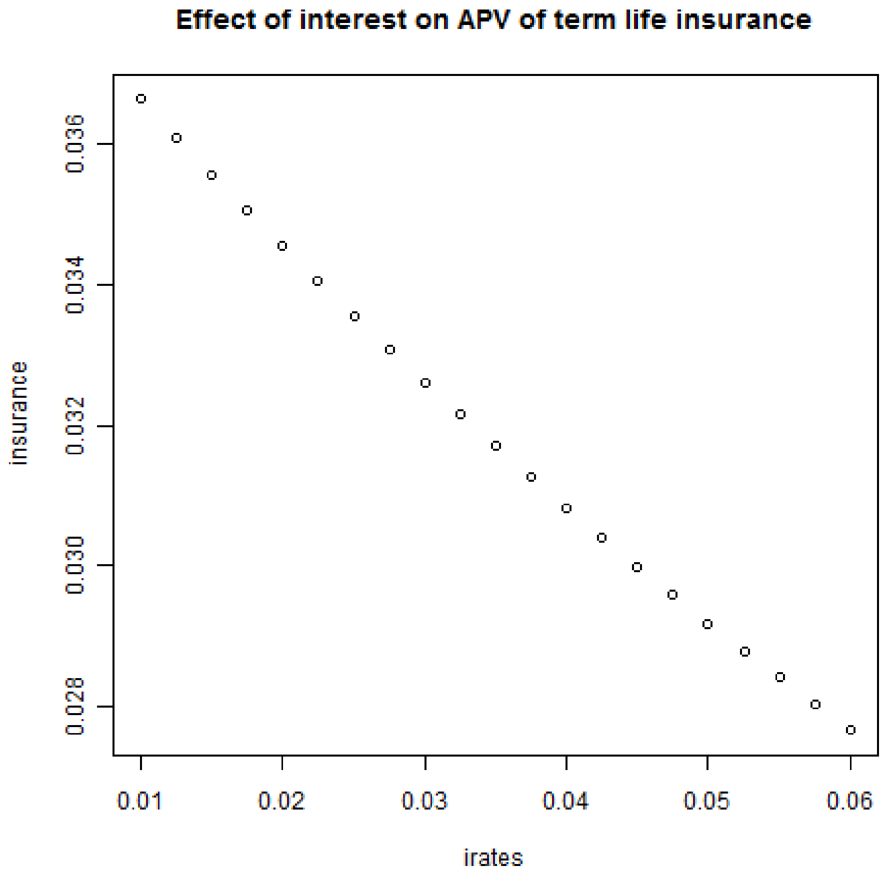

Figure 1: Interest rate effect on life insurance

Other examples

Figure 1 shows the effect of changing interest rates on the APV of $A^{\,1}_{40:\overline{10|}}$. The APV is a present value of a random variable that represent a composite function between the discount amount and indicator variables regarding the life status of the insured. Figure 2 shows the stochastic distribution of $\ddot{a}_{65}$.

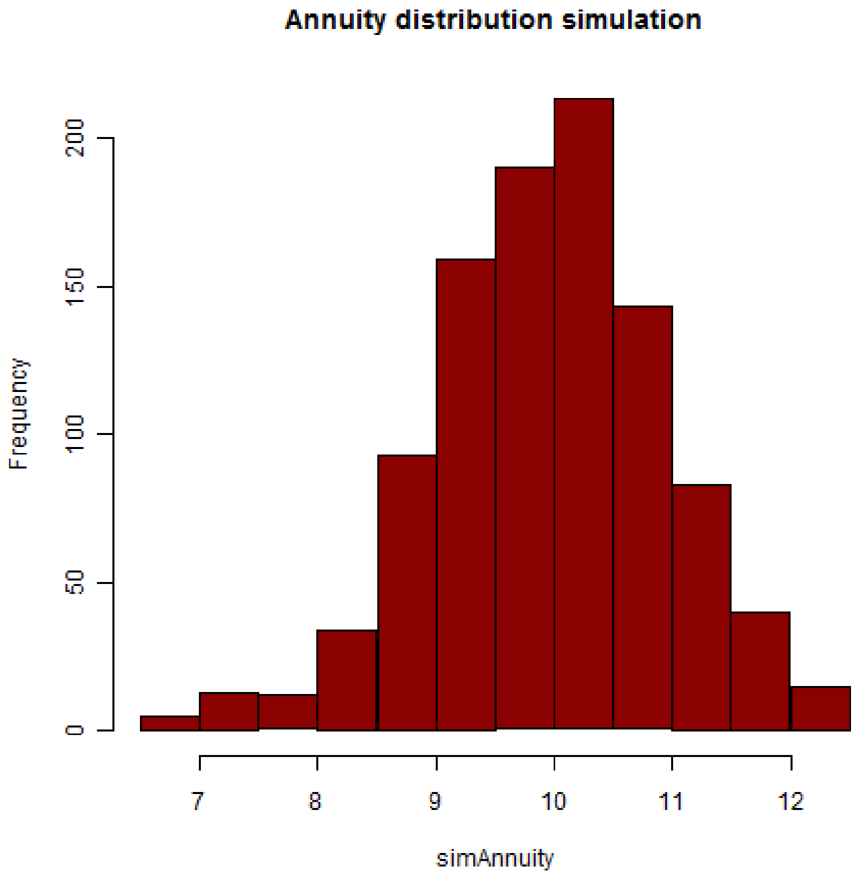

Figure 2: Stochastic distribution of $\ddot{a}_6 5$

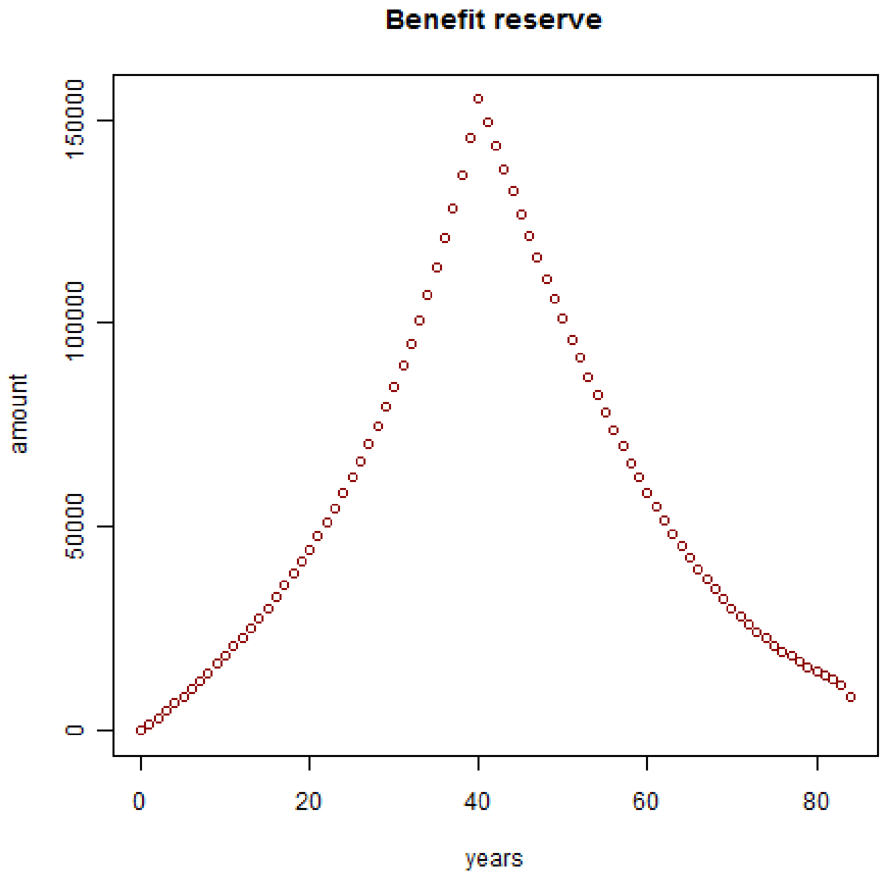

Figure 3: Benefit reserve of $\ddot{a}_{65}$

5. Discussion

Lifecontingencies package allows practictioner actuaries to evaluate actuarial present values functions by means of the R system framework. The lifecontingencies packages offers the basic tools to manipulate life tables, time value of cash flows. These tools are used to evaluate standard life contingencies present values by code already binded to the package as long as to build own function to perform day to day actuarial analysis.

Future work spans in multiple directions. Carefull check of the APV functions will be performed, expecially in the computation of stochastic values. C++ fragments will be tested and addedd to the package whether performance shows to improve.
Finally coerce functions will be written. We wish to provide input and output convenience functions for lifecontingencies objects toward package specialized in demographic analysis. Moreover the use of stochastic interest rate within the actuarial analysis will be facilitated allowing the package to interact with specialized packages.

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.
This 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

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.

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=PxSVxES7Sj0C.

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

Rigby RA, Stasinopoulos DM (2005). "Generalized additive models for location, scale and shape,(with discussion)." *Applied Statistics*, **54**, 507–554.

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