

# Intro to the lifecontingencies R package

Giorgio Alfredo Spedicato, Ph.D C.Stat ACAS

24 settembre, 2015



# Intro

- ▶ The lifecontingencies package (Spedicato 2013) will be introduced.

# Intro

- ▶ The lifecontingencies package (Spedicato 2013) will be introduced.
- ▶ As first half 2015 it is the first R (Team 2012) package merging demographic and financial mathematics function in order to perform actuarial evaluation of life contingent insurances (annuities, life insurances, endowments, etc).

# Intro

- ▶ The lifecontingencies package (Spedicato 2013) will be introduced.
- ▶ As first half 2015 it is the first R (Team 2012) package merging demographic and financial mathematics function in order to perform actuarial evaluation of life contingent insurances (annuities, life insurances, endowments, etc).
- ▶ The applied examples will shown: how to load the R package, how to perform basic financial mathematics and demographic calculations, how to price and reserve financial products.

- ▶ The final example will show how to mix lifecontingencies and demography (Rob J Hyndman et al. 2011) function to assess the mortality development impact on annuities.

- ▶ The final example will show how to mix lifecontingencies and demography (Rob J Hyndman et al. 2011) function to assess the mortality development impact on annuities.
- ▶ The interested readers are suggested to look to the package's vignettes (also appeared in the Journal of Statistical Software) for a broader overview. (Dickson, Hardy, and Waters 2009; and Mazzoleni 2000) provide and introduction of Actuarial Mathematics theory.

- ▶ The final example will show how to mix lifecontingencies and demography (Rob J Hyndman et al. 2011) function to assess the mortality development impact on annuities.
- ▶ The interested readers are suggested to look to the package's vignettes (also appeared in the Journal of Statistical Software) for a broader overview. (Dickson, Hardy, and Waters 2009; and Mazzoleni 2000) provide and introduction of Actuarial Mathematics theory.
- ▶ Also (Charpentier 2012) and (Charpentier 2014) discuss the software.



# First moves into the lifecontingencies package

## Loading the package

- ▶ The package is loaded using

```
library(lifecontingencies) #load the package
```

# First moves into the lifecontingencies package

## Loading the package

- ▶ The package is loaded using

```
library(lifecontingencies) #load the package
```

- ▶ It requires a recent version of R ( $\geq 3.0$ ) and the markovchain package (Spedicato, Giorgio Alfredo 2015). The development version of the package requires also Rcpp package (Eddelbuettel 2013).

## Financial mathematics

- ▶ Actuarial mathematics calculation applies probability to quantify uncertainty on present values calculation.

```
capitals <- c(-1000,200,500,700)
times <- c(0,1,2,5)
#calculate a present value
presentValue(cashFlows=capitals, timeIds=times,
             interestRates=0.03)
```

```
## [1] 269.2989
```

## Financial mathematics

- ▶ Actuarial mathematics calculation applies probability to quantify uncertainty on present values calculation.
- ▶ Certain present value calculations can be directly evaluated by the package

```
capitals <- c(-1000,200,500,700)
times <- c(0,1,2,5)
#calculate a present value
presentValue(cashFlows=capitals, timeIds=times,
             interestRates=0.03)
```

```
## [1] 269.2989
```

- ▶ The `presentValue` function is the kernel of functions that are used to calculate annuities and accumulated values, also paid in fractional  $k$  payments.

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

```
## [1] 4.579707 4.653791
```

- ▶ Such functions can be combined to price bonds and other classical financial products.

```
bondPrice<-5*annuity(i=0.03,n=10)+100*1.03^-10  
bondPrice
```

```
## [1] 117.0604
```

- ▶ Such functions can be combined to price bonds and other classical financial products.
- ▶ The following code exemplifies the calculation of a 5% coupon bond at 3% yield rate when the term is ten year.

```
bondPrice<-5*annuity(i=0.03,n=10)+100*1.03^-10  
bondPrice
```

```
## [1] 117.0604
```

# Managing lifetables and actuarial tables

- ▶ A lifetable object is an S4 class comprised by three slots: the age (from 0 to  $\omega$ ), the people at risk at beginning of age  $x$  and the table name.

```
#create an demo lifetable  
xDemo<-seq(from=0,to=5,by=1)  
lxDemo<-c(100,95,90,60,30,5)  
lifetableDemo<-new("lifetable",x=xDemo,  
                    lx=lxDemo,name="Demo")
```



- In practice, it is often more convenient to load an existing table from a CSV or XLS source. Some common tables have been bundled as data.frames within the package.

```
data(demoIta) #using the internal Italian LT data set
lxIPS55M <- with(demoIta, IPS55M)
#performing some fixings
pos2Remove <- which(lxIPS55M %in% c(0,NA))
lxIPS55M <- lxIPS55M[-pos2Remove]
xIPS55M <- seq(0, length(lxIPS55M)-1, 1)
#creating the table
ips55M <- new("lifetable", x=xIPS55M,
lx=lxIPS55M, name="IPS 55 Males")
```

- ▶ In practice, it is often more convenient to load an existing table from a CSV or XLS source. Some common tables have been bundled as data.frames within the package.
- ▶ The example that follows creates the Italian IPS55 life table.

```
data(demoIta) #using the internal Italian LT data set  
lxIPS55M <- with(demoIta, IPS55M)  
#performing some fixings  
pos2Remove <- which(lxIPS55M %in% c(0,NA))  
lxIPS55M <- lxIPS55M[-pos2Remove]  
xIPS55M <- seq(0, length(lxIPS55M)-1, 1)  
#creating the table  
ips55M <- new("lifetable", x=xIPS55M,  
lx=lxIPS55M, name="IPS 55 Males")
```

- ▶ It is therefore easy to perform standard demographic calculations with the aid of package functions.

```
# decrements between age 65 and 70  
dxt(ips55M, x = 65, t = 5)
```

```
## [1] 3659.74
```

```
# probabilities of death between age 80 and 85  
qxt(ips55M, x = 80, t = 2)
```

```
## [1] 0.07264833
```

```
# expected curtate lifetime  
exn(ips55M, x = 65)
```

```
## [1] 21.96873
```

# Pricing and reserving Life Contingent insurances

- ▶ The pricing and reserving of pure endowments  $A_{x:\overline{n}|}^1$ , term insurances,  $A_{x:\overline{n}|}^1$ , annuities  $\ddot{a}_x^{\{m\}}$ , endowments  $A_{x:\overline{n}|}^1$  can be easily handled within the package.

```
#creates a new actuarial table  
ips55Act<-new("actuarialtable",  
x=ips55M@x,lx=ips55M@lx,  
interest=0.02,name="IPS55M")
```

# Pricing and reserving Life Contingent insurances

- ▶ The pricing and reserving of pure endowments  $A_{x:\overline{n}|}^1$ , term insurances,  $A_{x:\overline{n}|}^1$ , annuities  $\ddot{a}_x^{\{m\}}$ , endowments  $A_{x:\overline{n}|}^1$  can be easily handled within the package.
- ▶ Performing such actuarial calculations requires to create an actuarial table (similar to a lifetable but with one more slot for the interest rate) that can be create as it is shown below.

```
#creates a new actuarial table  
ips55Act<-new("actuarialtable",  
x=ips55M@x,lx=ips55M@lx,  
interest=0.02,name="IPS55M")
```

- The example that follows computes the yearly premium  $P = 50000 * \frac{{}_{35}E_{30}}{{}_a{_{30:35}}}$  for a pure endowment on 50K capital.

```
#compute APV  
APV=50e3*Exn(actuarialtable =  
ips55Act,x=30,n=35)  
#compute Premium  
P=APV/axn(actuarialtable =  
ips55Act,x=30,n=35)  
c(APV,P)
```

```
## [1] 23584.7564    938.1422
```

- ▶ The package allows to easily perform sensitivities on the financials changing key parameters like term, interest, etc as we can see on a endowment purchased by a 30 years old policyholder for 30 years.

```
#defining the ranges
```

```
interest.range<-seq(from=0.015, to=0.035,by=0.001)
```

```
term.range<-seq(from=20, to=40,by=1)
```

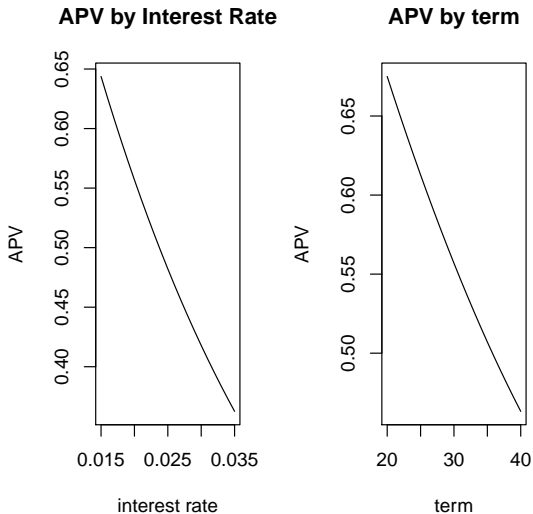
```
#computing APV sensitivities
```

```
apv.interest.sensitivity<-sapply(interest.range,
```

```
FUN = "AExn",actuarialtable=ips55Act,x=30,n=30)
```

```
apv.term.sensitivity<-sapply(term.range,FUN = "AExn",  
                             actuarialtable=ips55Act,x=30)
```

- ▶ The graph below displays sensitivities on APV varying interest rate and insurance term.





- ▶ Also, calculation of outstanding reserves is made easy as the following example, applied on a 100K face value 40 year term term life insurance with payments made yearly on a policyholder aged 25, shows.

*#compute the APV and premium*

```
APV=100e3*Axn(actuarialtable = ips55Act,x=25,n=40)
```

```
P=APV/axn(actuarialtable = ips55Act,x=25,n=40)
```

*#define a reserve function*

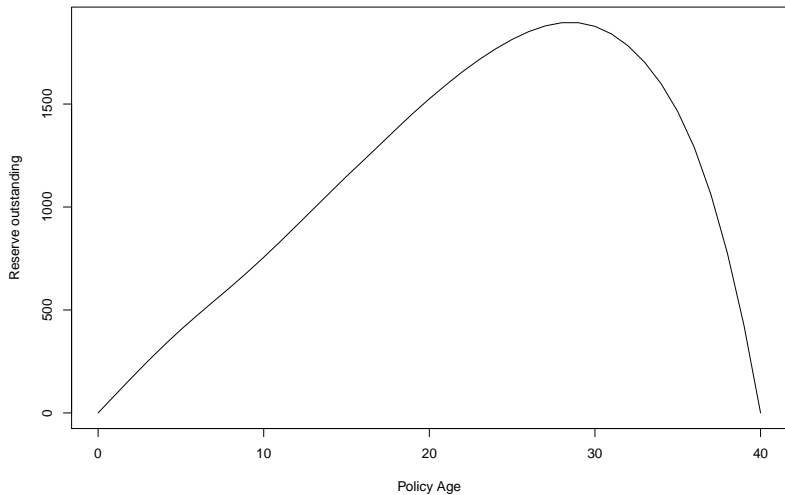
```
reserveFunction<-function(t)
```

```
  100e3*Axn(actuarialtable = ips55Act,x=25+t,n=40-t) -
```

```
  P *axn(actuarialtable = ips55Act,x=25+t,n=40-t)
```

```
reserve<-sapply(0:40,reserveFunction)
```

## Reserve



# Stochastic evaluation

- ▶ The APV of a life contingent insurance is the expected value of a function of a random variable (the residual life time). Knowing the interest rate and the underlying life table it is possible both to compute the APV and to assess the distribution of the underlying life insurance.

```
#analyzing and Endowment of 100K on x=40, n=25  
#compute APV  
APV=AExn(actuarialtable = ips55Act,x=40,n=25)  
#sampling  
AEXnDistr<-rLifeContingencies(n=10e3,  
lifecontingency = "AExn",x = 40,  
t=25,object = ips55Act)
```

- In order to assess if the distribution is unbiased we use a classical one sample t - test.

```
#assess if the expected value match the theoretical one  
t.test(x=AEXnDistr,mu = APV)
```

```
##
```

```
## One Sample t-test
```

```
##
```

```
## data: AEXnDistr
```

```
## t = 1.0199, df = 9999, p-value = 0.3078
```

```
## alternative hypothesis: true mean is not equal to 0.6148
```

```
## 95 percent confidence interval:
```

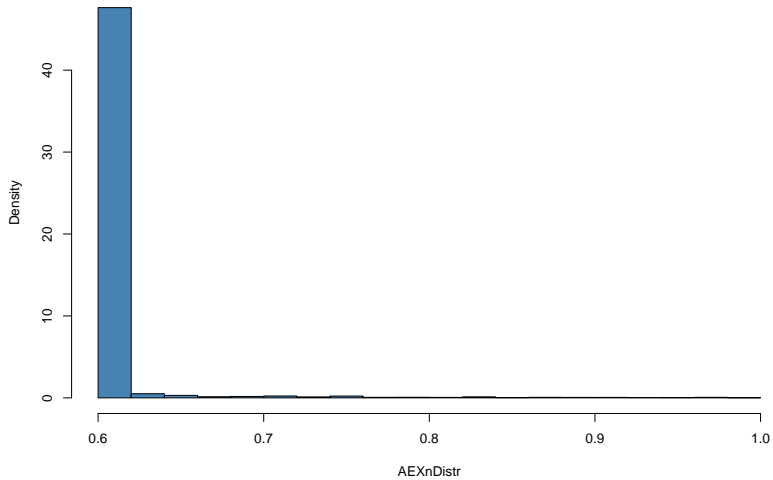
```
## 0.6145516 0.6158519
```

```
## sample estimates:
```

```
## mean of x
```

```
## 0.6152018
```

## Endowment Actuarial Value Distribution



# Assessing longevity impact on annuities using lifecontingencies and demography

- ▶ This part of the presentation will make use of the demography package to calibrate Lee Carter (Lee and Carter 1992) model,  $\log(\mu_{x,t}) = a_x + b_x * k_t \rightarrow p_{x,t} = \exp^{-\mu_{x,t}}$ , projecting mortality and implicit life tables.

```
#load the package and the italian tables  
library(demography)  
#italyDemo<-hmd.mx("ITA", username="yourUN",  
#password="yourPW")  
load(file="mortalityDatasets.RData") #load the dataset
```

- ▶ Lee Carter model is calibrated using lca function of demography package.

```
#calibrate lee carter  
italy.leecarter<-lca(data=italyDemo,series="total",  
                    max.age=103,adjust = "none")  
#perform modeling of kt series  
kt.model<-auto.arima(italy.leecarter$kt)  
#projecting the kt  
kt.forecast<-forecast(kt.model,h=100)
```

- ▶ Lee Carter model is calibrated using lca function of demography package.
- ▶ Then an arima model is used to project (extrapolate) the underlying  $k_t$  over the historical period.

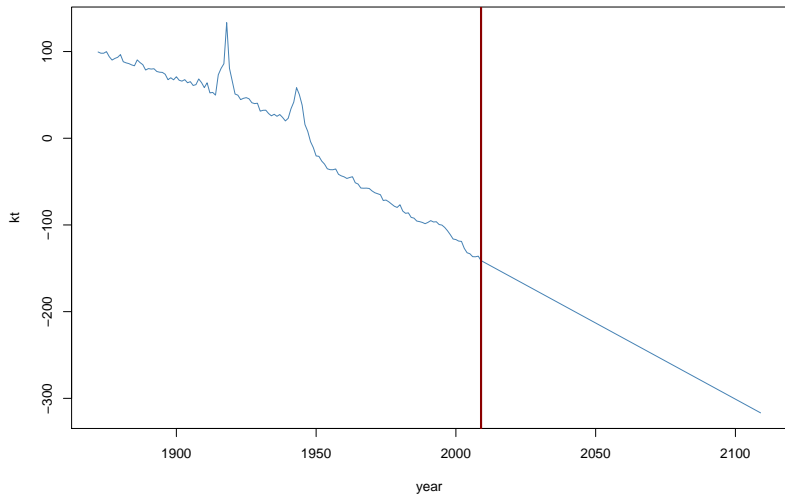
```
#calibrate lee carter  
italy.leecarter<-lca(data=italyDemo,series="total",  
                    max.age=103,adjust = "none")  
#perform modeling of kt series  
kt.model<-auto.arima(italy.leecarter$kt)  
#projecting the kt  
kt.forecast<-forecast(kt.model,h=100)
```



-The code below generates the matrix of prospective life tables

```
#indexing the kt
kt.full<-ts(union(italy.leecarter$kt, kt.forecast$mean),
            start=1872)
#getting and defining the life tables matrix
mortalityTable<-exp(italy.leecarter$ax
+italy.leecarter$bx%*%t(kt.full))
rownames(mortalityTable)<-seq(from=0, to=103)
colnames(mortalityTable)<-seq(from=1872,
to=1872+dim(mortalityTable)[2]-1)
```

historical and projected KT



- ▶ now we need a function that returns the one-year death probabilities given a year of birth (cohort.

```
getCohortQx<-function(yearOfBirth)
{
  colIndex<-which(colnames(mortalityTable)
                  ==yearOfBirth) #identify
  #the column corresponding to the cohort
  #define x the probabilities from which
  #the projection is to be taken
  maxLength<-min(nrow(mortalityTable)-1,
                 ncol(mortalityTable)-colIndex)
  qxOut<-numeric(maxLength+1)
  for(i in 0:maxLength)
    qxOut[i+1]<-mortalityTable[i+1,colIndex+i]
  #fix: we add a fictional omega age where
  #death probability = 1
  qxOut<-c(qxOut,1)
  return(qxOut)
}
```

- ▶ Now we use such function to obtain prospective life tables and to perform actuarial calculations. For example, we can compute the APV of an annuity on a workers' retiring at 65 assuming he were born in 1920, in 1950 and in 1980. We will use the interest rate of 1.5% (the one used to compute Italian Social Security annuity factors).

*#generate the life tables*

```
qx1920<-getCohortQx(yearOfBirth = 1920)
lt1920<-probs2lifetable(probs=qx1920,type="qx",
name="Table 1920")
at1920<-new("actuarialtable",x=lt1920@x,
lx=lt1920@lx,interest=0.015)
qx1950<-getCohortQx(yearOfBirth = 1950)
lt1950<-probs2lifetable(probs=qx1950,
type="qx",name="Table 1950")
at1950<-new("actuarialtable",x=lt1950@x,
lx=lt1950@lx,interest=0.015)
qx1980<-getCohortQx(yearOfBirth = 1980)
```

- ▶ Now we use such function to obtain prospective life tables and to perform actuarial calculations. For example, we can compute the APV of an annuity on a workers' retiring at 65 assuming he were born in 1920, in 1950 and in 1980. We will use the interest rate of 1.5% (the one used to compute Italian Social Security annuity factors).
- ▶ The first step is to generate the life and actuarial tables

*#generate the life tables*

```
qx1920<-getCohortQx(yearOfBirth = 1920)
lt1920<-probs2lifetable(probs=qx1920,type="qx",
name="Table 1920")
at1920<-new("actuarialtable",x=lt1920@x,
lx=lt1920@lx,interest=0.015)
qx1950<-getCohortQx(yearOfBirth = 1950)
lt1950<-probs2lifetable(probs=qx1950,
type="qx",name="Table 1950")
at1950<-new("actuarialtable",x=lt1950@x,
lx=lt1950@lx,interest=0.015)
qx1980<-getCohortQx(yearOfBirth = 1980)
```

- Now we can evaluate  $\ddot{a}_{65}$  and  $\ddot{e}_{65}$  for workers born in 1920, 1950 and 1980 respectively.

```
cat("Results for 1920 cohort","\n")
```

```
## Results for 1920 cohort
```

```
c(exn(at1920,x=65),axn(at1920,x=65))
```

```
## [1] 16.51391 15.14127
```

```
cat("Results for 1950 cohort","\n")
```

```
## Results for 1950 cohort
```

```
c(exn(at1950,x=65),axn(at1950,x=65))
```

```
## [1] 18.72669 16.83391
```

```
cat("Results for 1980 cohort","\n")
```

# Bibliography I

Charpentier, Arthur. 2012. “Actuarial Science with R 2: Life Insurance and Mortality Tables.”

<http://freakonometrics.blog.free.fr/index.php?post/2012/04/04/Life-insurance,-with-R,-Meielisalp>.

———. 2014. *Computational Actuarial Science*. The R Series. Cambridge University Press.

Dickson, D.C.M., M.R. Hardy, and H.R. Waters. 2009. *Actuarial Mathematics for Life Contingent Risks*. International Series on Actuarial Science. Cambridge University Press.

Eddelbuettel, Dirk. 2013. *Seamless R and C++ Integration with Rcpp*. New York: Springer.

Lee, R.D., and L.R. Carter. 1992. “Modeling and Forecasting U.S. Mortality.” *Journal of the American Statistical Association* 87 (419): 659–75. doi:10.2307/2290201.

## Bibliography II

Mazzoleni, P. 2000. *Appunti Di Matematica Attuariale*. EDUCatt Università Cattolica.

Rob J Hyndman, Heather Booth, Leonie Tickle, and John Maindonald. 2011. *Demography: Forecasting Mortality, Fertility, Migration and Population Data*.

<http://CRAN.R-project.org/package=demography>.

Spedicato, Giorgio Alfredo. 2013. "The lifecontingencies Package: Performing Financial and Actuarial Mathematics Calculations in R." *Journal of Statistical Software* 55 (10): 1–36.

<http://www.jstatsoft.org/v55/i10/>.

Spedicato, Giorgio Alfredo. 2015. *Markovchain: Discrete Time Markov Chains Made Easy*.

Team, R Development Core. 2012. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <http://www.R-project.org/>.