LTC insurance with markovchain

Giorgio A. Spedicato, Ph.D FCAS FSA

07 giugno, 2017

Disclaimer

The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of his Employer (the Unipol Group). The Unipol Group neither does not guarantee the accuracy or reliability of the information provided herein nor takes position on.

Intro

LTC insurance

- As well known, Long Term Care (LTC) policies guarantee annuity benefits as long as a disability status is present.
- The actuarial approach to model LTC insurances have been traditionally based on the multi - state approach.
- The demographic assumptions are represented by tables that model the transition between (A)ctive (healthy), disabled/(I)ll and (D)ead status across ages.

- The lump-sum premium for a yearly benefit C is $U = \sum_{h=1}^{\omega-x} p_{x,h-1}^{hh} * q_{x+h-1}^{hi} * v^h * \ddot{a}_{x+h}^{(i)} = P * \ddot{a}_x^{(h)}$.
- The annual premium is conventionally paid when the policyholder is (H)ealthy.
- Reserves depends by the attained status (H or I) at the evaluation period.

Empirical data

- We used assumptions for the Italian population taken from (Paolo de Angelis 2016).
- In particular, this exercise is based on the transition probabilities for Italian male population estimated for the 2016 calendar year.
- Possible states are (A)ctive, (I)ll and (D)ead. Modeled transitions are from A -> I, from A -> D and I -> D.

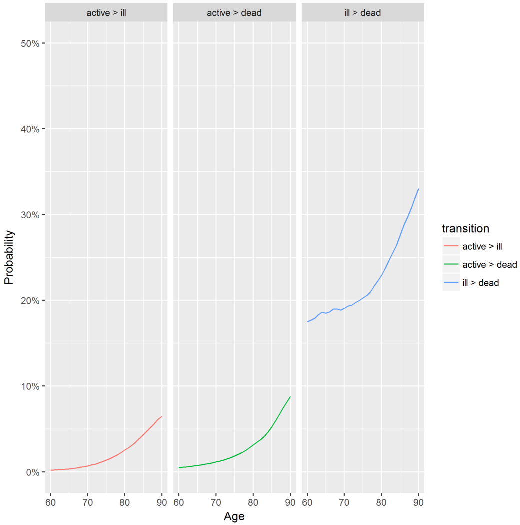


Figure 1: Italian Males Transition Probabilities 2016

The markovchain package

Purpose

- Package created for easily handling Discrete Time Markov Chains (DTMC) in R by S4 classes for homogeneous and not DTMCs.
- Also used to perform structural analysis (e.g. states classification), statistical inference, estimation and simulation.
- The functions written to perform the actuarial analysis on LTC data heavily relies on the markovchain package's simulation functions.

Package history

- On Cran since mid 2013.
- Development sponsored by Google within the Google Summer of Code 2015, 2016 and 2017.
- Core parts written in Rcpp (Eddelbuettel 2013). The simulation function also uses RcppParallel (Allaire et al. 2016).

Application to LTC insurance

- The stochastic process underlying a LTC insurance can be considered a non - homogeneous DTMC, since transition probabilities vary by age.

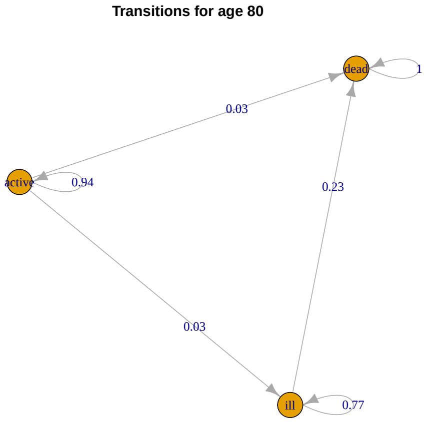
```
#defining the transition diagram for age 80
mc80<-createAgeMarkovChain(80)
mc80
```

```
80
 A  3 - dimensional discrete Markov Chain defined by the following states:
 active, ill, dead
 The transition matrix  (by rows)  is defined as follows:
          active        ill       dead
active 0.9432764 0.02541479 0.03130879
ill    0.0000000 0.77097391 0.22902609
dead   0.0000000 0.00000000 1.00000000
```

```r
#summarizing the structural proprieties of
#the transition diagram
summary(mc80)
```

```
80  Markov chain that is composed by:
Closed classes:
dead
Recurrent classes:
{dead}
Transient classes:
{active},{ill}
The Markov chain is not irreducible
The absorbing states are: dead
```

```r
#plotting
plot(mc80,main="Transitions for age 80")
```



- The markovchain package allows to draw samples from non - homogenous DTMC.

```
# simulating life trajectories
table90 <- getTable(age = 90)
simulateLifeTrajectiories(transitionTable = table90, numSim =
```

```
  91        92        93     94     95     96     97     98     99
1 "active" "active" "ill" "ill" "ill" "dead" "dead" "dead" "de
  101    102    103    104    105    106    107    108    109
1 "dead" "dead" "dead" "dead" "dead" "dead" "dead" "dead" "dea
  111    112    113    114    115    116    117    118    119
1 "dead" "dead" "dead" "dead" "dead" "dead" "dead" "dead" "dea
  121    122
1 "dead" "dead"
```

Actuarial analysis of LTC coverages with R

Assumptions

- A simulation approach is used to actuarially evaluate the LTC coverage.
- The (fictionary) policyholder age is 75, the real interest rate is $i = 0.01$.
- we will compute benefit premiums (lump sum and yearly) and reserves.

- A large number of random life trajectories is sampled.
- Cash flows depends by the status at the beginning of the year, 1 when (D)isabled, 0 otherwise.
- The yearly benefit is set to 12K.

Simulating life trajectories

```
## retrieving transition since age 75
table.75<-getTable(75)
## simulating life trajectories
lifetrajectories.75<-simulateLifeTrajectiories(transitionTable
  numSim = 1000,
  include_start_age = FALSE,
  begin_status = "active")
```

```r
#sampled life trajectories for a policyholder aged 75
lifetrajectories.75[1:5,10:15]
```

```
  85       86       87       88       89       90
1 "dead"   "dead"   "dead"   "dead"   "dead"   "dead"
2 "active" "dead"   "dead"   "dead"   "dead"   "dead"
3 "dead"   "dead"   "dead"   "dead"   "dead"   "dead"
4 "active" "active" "active" "active" "active" "active"
5 "active" "active" "active" "active" "active" "active"
```

```r
#computing expected future years disabled
disabled01<-matrix(0,nrow=nrow(lifetrajectories.75),
ncol=ncol(lifetrajectories.75))
disabled01[which(lifetrajectories.75=="ill")]=1
mean(rowSums(disabled01))
```

```
[1] 1.313
```

Computing premiums

```
#PV of a policy paying 12K euro at the beginning of the year
#if the policyholder is disabled
##simulating
pvbenefits.75<-getPVDistribution(lifetrajectoriesMatrix
= lifetrajectories.75, target = "ill",CF = 1000*12,
real_interest_rate = 0.01,
begin = 1)
```

```r
## computing APV of lump sum benefits
U<-mean(pvbenefits.75); U
```

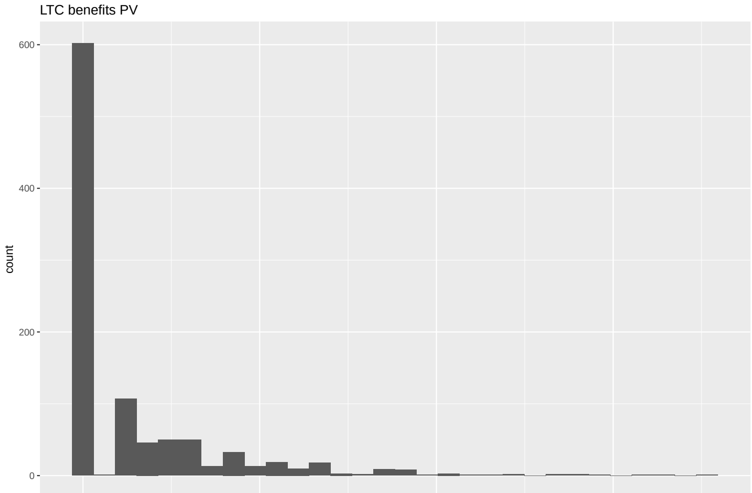
```
[1] 13983.72
```

```r
## yearly premium
annuity.75.healthy<-mean(getPVDistribution(lifetrajectoriesMatrix =
lifetrajectories.75, target = "active", CF = 1, real_interest_rate = 0.01,
P <- U/ annuity.75.healthy;P
```

```
[1] 1383.38
```

- It is also possible to perform a stochastic analysis of the future benefits distribution

```
qplot(pvbenefits.75,main = "LTC benefits PV",xlab="Euros")
```



Computing reserves

- A prospective approach is used.
- Reserves depend by attained status (Healthy or Ill)
- Calculations performed at example age of 80.

```
## retrieving transition from age 80
table.80<-getTable(80)
## simulating life trajectories (for healthy insureds)
lifetrajectories.80.healthy<-simulateLifeTrajectiories(transitionTable = ta
  numSim = 1000,
  include_start_age = FALSE,
  begin_status = "active")
## (for ill insureds)
lifetrajectories.80.ill<-simulateLifeTrajectiories(transitionTable = table.
  numSim = 1000,
  include_start_age = FALSE,
  begin_status = "ill")
```

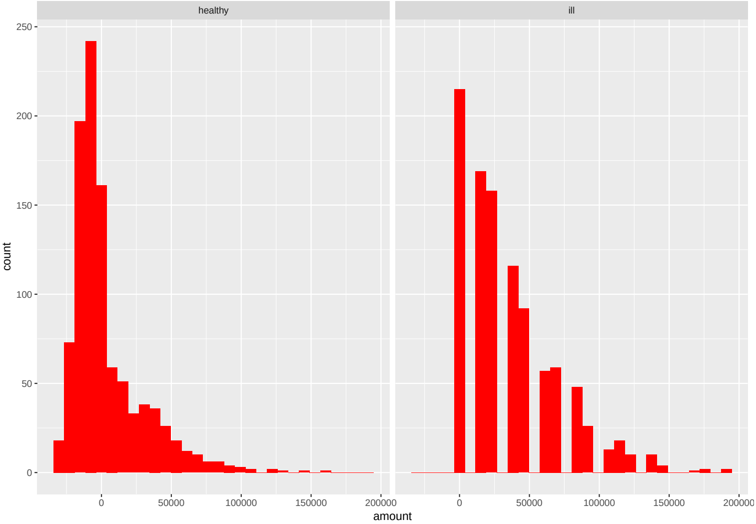
```
## benefit reserve distribution, 80 yo healthy
reserve.80.active.distr<-
  getReserveDistribution(lifetrajectoriesMatrix = lifetrajectories.80.healt
  CF_active = -P,CF_ill = +12000,CF_dead = 0,real_interest_rate = 0.01,begi
mean(reserve.80.active.distr)
```

```
[1] 3774.539
```

```
## benefit reserve distribution, 80 yo ill
reserve.80.ill.distr<-
  getReserveDistribution(lifetrajectoriesMatrix = lifetrajectories.80.ill,
  CF_active = -P,CF_ill = +12000,CF_dead = 0,real_interest_rate = 0.01,begi
mean(reserve.80.ill.distr)
```

```
[1] 35636.3
```

- It is also possible to quantify reserves' variability



References

Allaire, JJ, Romain Francois, Kevin Ushey, Gregory Vandenbrouck, Marcus Geelnard, and Intel. 2016. *RcppParallel: Parallel Programming Tools for 'Rcpp'*. `https://CRAN.R-project.org/package=RcppParallel`.

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

Paolo de Angelis, Luigi di Falco. 2016. *Assicurazioni Sulla Salute: Caratteristiche, Modelli E Basi Tecniche*. Il mulino.