Multiple decrements tables with lifecontingencies package

Giorgio Alfredo Spedicato, ACAS

Abstract

This paper introduces the `mdt` class within **lifecontingencies** R package, that handles multiple decrements models. Applied examples will show how perform demographic and actuarial calculation with the package

Keywords: actuarial mathematics, multiple decrement models, **lifecontingencies**.

1. Introduction

As of 2014 no R package provides easy tools to manage multiple decrement tables. Multiple decrement tables are the basis of many applications, for example in demography, medicine and actuarial science.
Until now no R package provides a good tool to manage multiple decrement tables, even if Deshmukh (2012) provides an R based focus on multiple decrement tables with applications in R.

This paper introduces the `mdt` class that has been specifically engineered to manage multiple decrements models with R. Applied examples will follows.

2. The mdt class

Examples in this paper are worked on slides provided in Valdez (2011). First of all, we load the R package.

```
R> library(lifecontingencies)
```

Then we create a `mdt` class object. We can use the first example found on (?, p. 4).

```
R> valdezDf<-data.frame(
+                    x=c(50:54),
+                    lx=c(4832555,4821937,4810206,4797185,4782737),
+                    hearth=c(5168, 5363, 5618, 5929, 6277),
+                    accidents=c(1157, 1206, 1443, 1679,2152),
+                    other=c(4293,5162,5960,6840,7631)
```

```
+  )
R> valdezMdt<-new("mdt",name="ValdezExample",table=valdezDf)

Added fictional decrement below last x and completed x and lx until zero....
Completed the table at top, all decrements on first cause
```

The mdt class is an S4 class object (Chambers 2008) comprised by a character slot `name` and a `data.frame` slot `table` that is composed by following columns:

1. `x`: the age, from 0 to ω.
2. `lx`: the subject living (at risk) at the beginning of age.
3. one or more colums for different causes of decrements.

Values within `table` item represents absolute number of subjects at risk or dying.

Within the various methods defined within the `mdt` class, `setValidity` performs consistency checks to properly create the `mdt` object. In particular, it verifies whether:

1. `x` and `lx` exist and that they are consistent. `x` should start from 0 and flows by increments of one. The first `lx` value should be equal to the sum of all decrements and that $l_x = l_{x-1} - (d_{x-1,1} + d_{x-1,2} + \ldots + d_{x-1,k})$ for any x.
2. If the decrements (or x and lx) have been provided only for partial ages, the table is completed below (from 0 to l_{x-1}) assuming a decrement rate of 0.01 for the first cause of death.
3. if the decrements at last provided age, ω, do not sum to l_ω, the table is incremented by one row such as $lx_{\omega+1} = lx_\omega - (d_{\omega,1} + d_{\omega,2} + \ldots + d_{\omega,j})$.

As shown, when the table is sanitized the operations performed are reported on logs.

An internal function, `.tableSanitizer` tries to fix the limitations on the input table in order it to meet the class definition requirements.

Table can be viewed thanks to a `print` and `show` method. Similarly, it is possible to export a `mdt` to a `data.frame` or to a `markovchainList` object (from **markovchain** package).

```
R> print(valdezMdt)

Multiple decrements table ValdezExample
       hearth    accidents       other
0  0.010000000 0.0000000000 0.000000000
1  0.010000000 0.0000000000 0.000000000
2  0.010000000 0.0000000000 0.000000000
3  0.010000000 0.0000000000 0.000000000
4  0.010000000 0.0000000000 0.000000000
5  0.010000000 0.0000000000 0.000000000
6  0.010000000 0.0000000000 0.000000000
```

7	0.010000000	0.0000000000	0.000000000
8	0.010000000	0.0000000000	0.000000000
9	0.010000000	0.0000000000	0.000000000
10	0.010000000	0.0000000000	0.000000000
11	0.010000000	0.0000000000	0.000000000
12	0.010000000	0.0000000000	0.000000000
13	0.010000000	0.0000000000	0.000000000
14	0.010000000	0.0000000000	0.000000000
15	0.010000000	0.0000000000	0.000000000
16	0.010000000	0.0000000000	0.000000000
17	0.010000000	0.0000000000	0.000000000
18	0.010000000	0.0000000000	0.000000000
19	0.010000000	0.0000000000	0.000000000
20	0.010000000	0.0000000000	0.000000000
21	0.010000000	0.0000000000	0.000000000
22	0.010000000	0.0000000000	0.000000000
23	0.010000000	0.0000000000	0.000000000
24	0.010000000	0.0000000000	0.000000000
25	0.010000000	0.0000000000	0.000000000
26	0.010000000	0.0000000000	0.000000000
27	0.010000000	0.0000000000	0.000000000
28	0.010000000	0.0000000000	0.000000000
29	0.010000000	0.0000000000	0.000000000
30	0.010000000	0.0000000000	0.000000000
31	0.010000000	0.0000000000	0.000000000
32	0.010000000	0.0000000000	0.000000000
33	0.010000000	0.0000000000	0.000000000
34	0.010000000	0.0000000000	0.000000000
35	0.010000000	0.0000000000	0.000000000
36	0.010000000	0.0000000000	0.000000000
37	0.010000000	0.0000000000	0.000000000
38	0.010000000	0.0000000000	0.000000000
39	0.010000000	0.0000000000	0.000000000
40	0.010000000	0.0000000000	0.000000000
41	0.010000000	0.0000000000	0.000000000
42	0.010000000	0.0000000000	0.000000000
43	0.010000000	0.0000000000	0.000000000
44	0.010000000	0.0000000000	0.000000000
45	0.010000000	0.0000000000	0.000000000
46	0.010000000	0.0000000000	0.000000000
47	0.010000000	0.0000000000	0.000000000
48	0.010000000	0.0000000000	0.000000000
49	0.010000000	0.0000000000	0.000000000
50	0.001069414	0.0002394179	0.000888350
51	0.001112209	0.0002501070	0.001070524
52	0.001167933	0.0002999872	0.001239032
53	0.001235933	0.0003499969	0.001425836

```
54 0.001312428 0.0004499516 0.001595530
55 1.000000000 0.0000000000 0.000000000
```

```
R> valdezDf<-as(valdezMdt,"data.frame")
R> valdezMarkovChainList<-as(valdezMdt,"markovchainList")
R>
```

Two specific methods have been defined for `mdt` class objects: `getOmega`, that returns the maximum attainable age (similar to the one of `lifetable` class), and `getDecrements`, that returns the decrements (by means of the names within table slot different from x and lx).

```
[1] 55
```

```
[1] "hearth"    "accidents" "other"
```

3. Decrement probabilities calculation

The **lifecontingencies** package makes easy to compute $d_x^{(j)}$, ${}_nd_x^{(j)}$ as well as ${}_nd_x^{(\tau)}$ quantities thanks to `dxt` function.

```
R> dxt(valdezMdt,x=51,decrement="other")
```

```
[1] 5162
```

```
R> dxt(valdezMdt,x=51,t=2, decrement="other")
```

```
[1] 11122
```

```
R> dxt(valdezMdt,x=51)
```

```
[1] 11731
```

Probabilities could be computed as well.

```
R> dxt(valdezMdt,x=51,t=2, decrement="other")
```

```
[1] 11122
```

```
R> pxt(valdezMdt,x=50,t=3)
```

```
[1] 0.9926809
```

```
R> qxt(valdezMdt,x=53,t=2,decrement=1)
```

```
[1] 0.002544409
```

4. Actuarial Applications

TO BE WRITTEN

Acknowledgments

TBD.

References

Chambers J (2008). *Software for Data Analysis: Programming with R*. Statistics and computing. Springer-Verlag. ISBN 9780387759357.

Deshmukh S (2012). *Multiple Decrement Models in Insurance: An Introduction Using R*. SpringerLink : Bücher. Springer. ISBN 9788132206590.

Valdez E (2011). "Multiple Decrement Models: Math 3631 Actuarial Mathematics II." http://www.math.uconn.edu/~valdez/math3631s11/M3631Weeks7to8-S2011-annot.pdf. Accessed: 25/04/2014.

Affiliation:

Giorgio Alfredo Spedicato
Ph.D ACAS C.STAT
Via Firenze 11 20037 Italy
E-mail: lifecontingencies@statisticaladvisor.com
URL: www.statisticaladvisor.com