Weighted decision trees applied to reserving in insurance

EAJ Conference - ISFA Lyon, LSAF
2016, 6-8 september

Xavier Milhaud[1]
Joint work with O. Lopez[2] and P. Thérond[1,3]

[1] LSAF, ISFA, Université Claude Bernard Lyon 1
[2] LSTA, Univ. Pierre et Marie Curie
[3] Galéa & Associés

Two possible approaches in reserving in insurance

There basically exist two ways of computing reserves :

1. Aggregated models (most famous : **Chain Ladder**) :
   - We work on C_{ij} : cumulated claim amounts reported in year i for development time j.
   - Underlying assumption : stationarity.
2. Claim by claim estimation :
   - Use of claims' characteristics to estimate the final claim amount.
   - Anticipate not yet reported claims.

Advantages and drawbacks

- Chain Ladder :
  - pros : compact data, easy to implement and understand,
  - cons : do not use (precise) information on claims, (very) strong underlying assumption...

- Micro-level reserving :
  - pros : use available information on claims,
  - cons : more difficult to put in practice, do not currently use the lifetime of claims...

Our approach : use all the information in the data

- Deal with the heterogeneity of data :
  - coming from their development time,
  - the claims' characteristics, ...
- Use Machine Learning techniques (data-driven) :
  - we focus on a weighted CART algorithm (decision tree),
  - implemented it and incorporated random forests extensions ;
- Be based on theoretical results :
  - see [LopezMilhaudTherond2016] : *Tree-based censored regression with applications in insurance*, to appear in the `Electronic Journal of Statistics`.

Our data

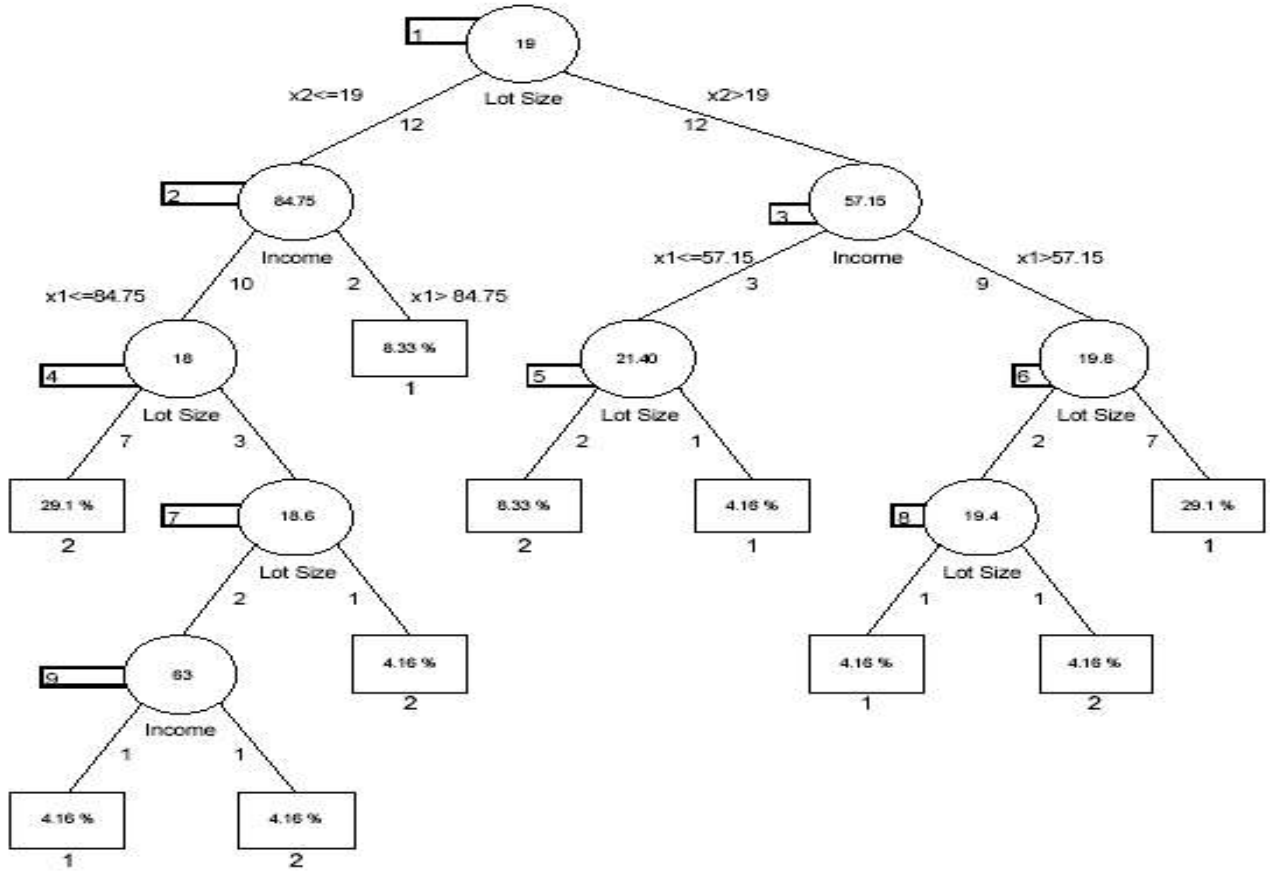
```
> ## Import database on which to work
> myData <- read.csv("myData.csv", header=TRUE)
> dim(myData)
[1] 83547    20

> summary(myData)
 Sex        TypeEmployee  ContractNumber          TYPE_ARRET        SurvDate
 F:65557   CAD: 3074     0725235: 1524   Maladie         :71563   Min.   :2006-01-0
 M:17990   ENP: 5879     0J98706:  879   Acc. Travail    :10644   1st Qu.:2007-05-1
           ETA:  713     0232097:  684   Maladie Hospi.. : 1035   Median :2008-08-2
           NCA:73290     0237127:  591   Maternite       :  179   Mean   :2008-07-2
           TNS:  591     0184638:  553   Longue Maladie  :   54   3rd Qu.:2009-10-2
                         0448817:  530   Maladie Serv.   :   23   Max.   :2010-11-3
                         (Other):78786   (Other)         :   49
    EndObsW          NonCensure          SPC         BegAgeClass BegAgeClassT
 Min.   :   1.00   Mode :logical   Employee:79882   1:21563     1:18685
 1st Qu.:  15.00   FALSE:5991      Manager : 3074   2:19039     2:11014
 Median :  42.00   TRUE :77556     Misc    :  591   3:20496     3:14589
 Mean   :  99.98   NA's :0                          4:22449     4:15570
 3rd Qu.: 106.00                                               5:23689
 Max.   :1578.00
```

	Sex	TypeEmployee	ContractNumber	TYPE_ARRET	SurvDate	BegIndDate	EndIndDat
23	F	NCA	0154496	Acc. Travail	2010-10-12	2010-11-11	2011-01-3
24	F	NCA	0154509	Maladie	2009-09-14	2009-10-14	2011-02-2
33	F	NCA	0154670	Maladie	2010-02-11	2010-03-13	2011-09-3
44	F	NCA	0156555	Maladie	2010-08-24	2010-09-23	2011-04-1
62	F	NCA	0161383	Maladie	2010-03-19	2010-04-18	2012-02-2
68	F	NCA	0161581	Maladie	2010-11-09	2010-12-09	2012-06-2
88	F	NCA	0331202	Maladie	2010-02-12	2010-03-14	2011-04-3
103	F	NCA	0385996	Maladie	2010-11-10	2010-12-10	2012-06-2
136	F	ENP	0725234	Maladie	2010-01-11	2010-02-10	2012-07-1
140	F	ENP	0725235	Maladie	2010-08-23	2010-09-22	2011-01-0

Cause	ComNet	BegAnc	EndAncInd	BegAge	EndObsW	NonCensure	SPC
Accident	Net_C	0	80	47.29363	50	FALSE	Employee
Sickness	Net_C	3	470	41.81246	443	FALSE	Employee
Sickness	Net_A	3	320	39.40041	293	FALSE	Employee
Sickness	Net_A	3	126	50.62286	99	FALSE	Employee
Sickness	Net_C	3	284	46.41752	257	FALSE	Employee
Sickness	Net_A	3	49	51.05544	22	FALSE	Employee
Sickness	Net_C	24	298	52.73374	292	FALSE	Employee
Sickness	Net_A	26	25	45.89733	21	FALSE	Employee
Sickness	Net_C	3	351	51.79466	324	FALSE	Employee
Sickness	Net_A	3	127	54.63107	100	FALSE	Employee

What is a decision tree ?

Goal : partition the population into different risk classes (classify).
→ The result provides with homogenous risk classes in terms of the variable of interest (response, i.e. claim amount).



Problem of censored claim amounts for reserving

Goal : estimate individual claim amounts T given features $\mathbf{X}$.

- Only observe the "follow-up" current claim amount Y : potentially censored observation.

If censored :

- The claim is still opened and has been under payment for some time (the claim is not closed).
- The total claim amount T is still unknown : just paid $Y \leq T$.

How to deal with non-closed claims ?

- A BAD solution : use only the fully settled (closed) claims to build the decision tree and estimate the quantity of interest.
  $\Rightarrow$ Selection bias : the characteristics of these claims are over-represented, in particular (short) development times.
  $\rightarrow$ **Tendency to underestimate the final claim amounts and thus the reserve**.

- However, the still in-payment claims provides a biased information $\Rightarrow$ necessity to correct this bias.

- Solution : overweight the (closed) claims with long development times to compensate their under-representativity
  $\Rightarrow$ use the non-closed claims and other information to compute these weights (Kaplan-Meier weights).

How to proceed in practice to compare the effectiveness of reserving methods ?

Use backtesting ! Prepare the data like this :

1. consider only closed claims, so that you know the full information about the final claim amount ;
2. introduce a censoring mechanism to artificially define still-in-payment claims, where the censoring mechanism leads to a censoring rate roughly similar to the original database ;
3. define learning and test samples :
   - learning sample : to build the censored regression tree,
   - validation sample : to compare the predictions of reserves given by the tree on this new dataset and observations.
4. estimate the parameters of Mack model on the validation sample et compute the reserves ;
5. compare errors !

Final database and results

```
> cat(round((sum(myData[ ,which(names(myData) == "NonCensure")] == FALSE) / nrow
7.17%

> head(myData, n = 6)
  Sex TypeEmployee ContractNumber TYPE_ARRET   SurvDate BegIndDate EndIndDate
1   F          NCA        0001591    Maladie 2007-11-03 2007-12-03 2007-12-21 Si
2   F          NCA        0001591    Maladie 2008-02-04 2008-03-05 2008-08-31 Si
3   M          NCA        0006192    Maladie 2006-12-24 2007-01-23 2007-04-30 Si
4   M          NCA        0006192    Maladie 2009-11-18 2009-12-18 2010-10-01 Si
5   F          NCA        0024191    Maladie 2006-03-20 2006-04-19 2006-09-03 Si
6   F          NCA        0024251    Maladie 2008-06-21 2008-07-21 2010-07-31 Si

  X2006.10.01 X2007.01.01 X2007.04.01 X2007.07.01 X2007.10.01 X2008.01.01 X2008.
1          NA          NA          NA          NA    17.99769          NA
2          NA          NA          NA          NA          NA     91.3125    87.
3          NA     91.3125    5.688769          NA          NA          NA
4          NA          NA          NA          NA          NA          NA
5          NA          NA          NA          NA          NA          NA
6          NA          NA          NA          NA          NA          NA
```

```
> triangle.retenu # triangle non-cumulé
```

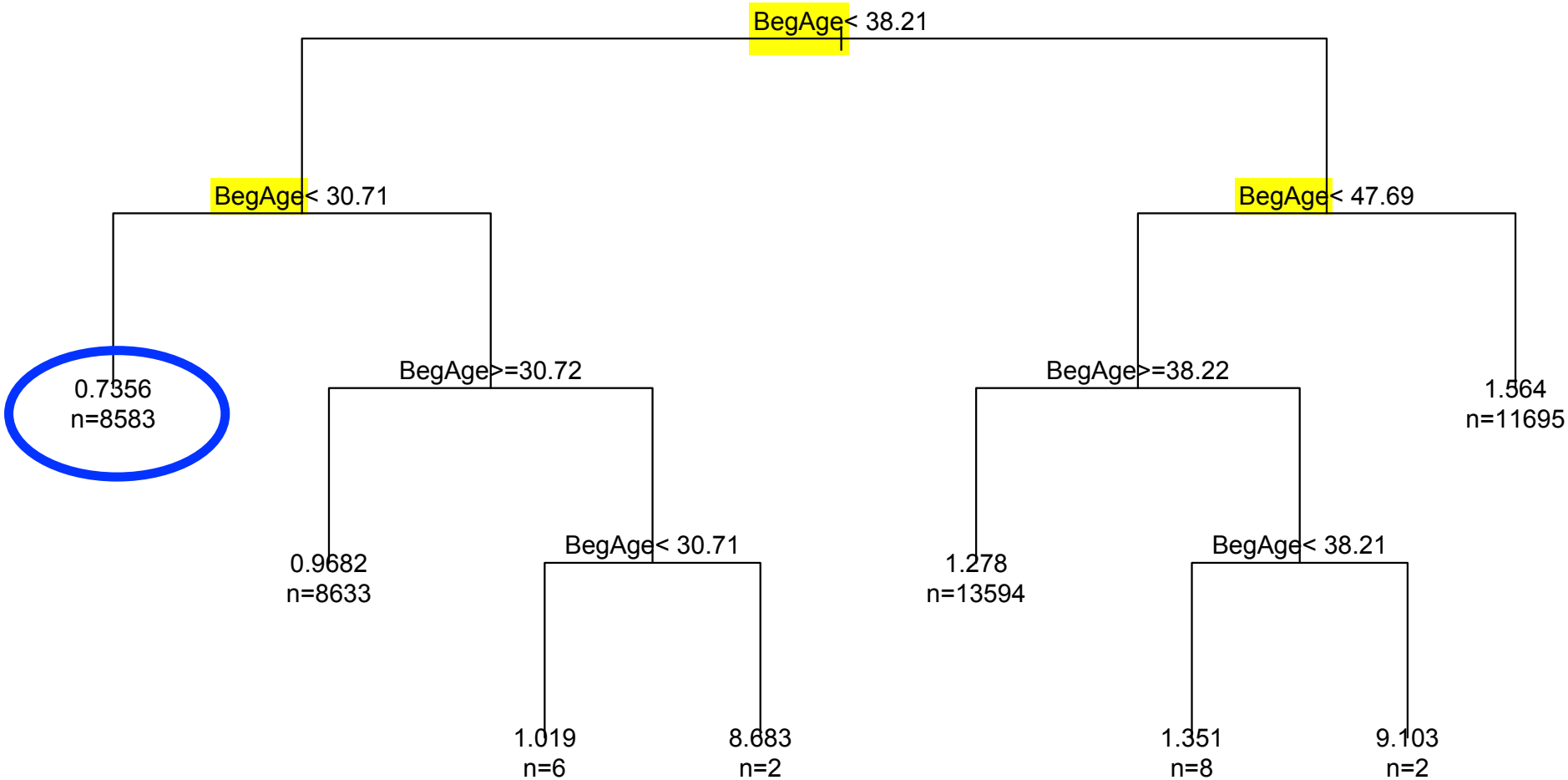
	dev1	dev2	dev3	dev4	dev5	dev6	dev7	dev8	dev9	dev10	dev11	dev12	de
2006-01-01	44860	17651	10234	7544	5604	4444	3295	2723	2152	1569	1146	473	
2006-04-01	55982	20923	13613	10572	7773	6206	5012	3792	2952	2520	1911	1009	
2006-07-01	49982	21727	13084	9081	6901	5749	4315	3572	3096	2277	1451	706	
2006-10-01	71692	29979	18480	13664	9214	6394	5281	4139	3045	2209	1705	745	
2007-01-01	63976	25548	15355	11246	7761	6178	5300	4291	3484	2586	1730	519	
2007-04-01	62908	24830	14731	11040	8339	6300	4817	3800	3373	3041	1241	NA	
2007-07-01	57010	24116	15602	12299	9915	7538	6190	4879	3844	1373	NA	NA	
2007-10-01	73432	28803	17001	12621	9621	7664	6332	4788	2112	NA	NA	NA	
2008-01-01	69086	26562	16313	11617	8293	6543	5151	2401	NA	NA	NA	NA	
2008-04-01	67486	26014	15696	10969	7681	5688	2287	NA	NA	NA	NA	NA	
2008-07-01	62748	25840	13842	9859	7388	3051	NA	NA	NA	NA	NA	NA	
2008-10-01	77569	29532	17040	11906	4445	NA	NA	NA	NA	NA	NA	NA	
2009-01-01	66986	25893	14549	5022	NA	NA	NA	NA	NA	NA	NA	NA	
2009-04-01	69909	26372	8442	NA	NA	NA	NA	NA	NA	NA	NA	NA	
2009-07-01	58504	12108	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
2009-10-01	45583	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	

```
> dim(learning.sample)
[1] 42523    37
```

```
> head(learning.sample)

   Sex TypeEmployee ContractNumber     TYPE_ARRET   SurvDate BegIndDate EndIndDa
1    F          NCA        0001591        Maladie 2007-11-03 2007-12-03 2007-12-
2    F          NCA        0001591        Maladie 2008-02-04 2008-03-05 2008-08-
3    M          NCA        0006192        Maladie 2006-12-24 2007-01-23 2007-04-
5    F          NCA        0024191        Maladie 2006-03-20 2006-04-19 2006-09-
9    F          NCA        0038268        Maladie 2006-05-02 2006-06-01 2006-07-
10   M          NCA        0064365 Maladie Hospi. 2006-10-30 2006-11-29 2007-02-

Cause ComNet BegAnc EndAncInd   BegAge EndObsW NonCensure      SPC
Sickness  Net_C      3        45 47.69884  0.1971       TRUE Employee
Sickness  Net_C      3       206 47.43053  1.9603       TRUE Employee
Sickness  Net_C      3       124 46.06982  1.0623       TRUE Employee
Sickness  Net_A     30       137 43.63313  1.5003       TRUE Employee
Sickness  Net_A     30        32 35.49897  0.3504       TRUE Employee
Sickness  Net_A      3       107 37.32786  0.8761       TRUE Employee


> KM.weights <- unlist(aft.kmweight(Y = matrix(data=learning.sample$EndObsW, nro
> sum(KM.weights)
[1] 1
```

```
> head(learning.sample)

    Sex TypeEmployee ContractNumber TYPE_ARRET   SurvDate BegIndDate EndIndDate
35    F          NCA        0154699    Maladie 2008-04-10 2008-05-10 2008-05-11
173   F          ENP        0729486    Maladie 2006-02-27 2006-03-29 2006-03-30
240   F          NCA        0149036    Maladie 2006-05-18 2006-06-17 2006-06-18
295   F          NCA        0637995    Maladie 2006-06-12 2006-07-12 2006-07-13
299   F          NCA        0637995    Maladie 2007-12-12 2008-01-11 2008-01-12
468   F          NCA        0179261    Maladie 2007-02-01 2007-03-03 2007-03-04

Cause ComNet BegAnc EndAncInd   BegAge EndObsW NonCensure      SPC    KM.weight
Sickness  Net_C      3        28 50.54346   0.011       TRUE Employee 2.351669e-
Sickness  Net_A      3        28 39.60849   0.011       TRUE Employee 2.351669e-
Sickness  Net_A      3        28 54.24778   0.011       TRUE Employee 2.351669e-
Sickness  Net_B      1        30 52.67077   0.011       TRUE Employee 2.351669e-
Sickness  Net_B      1        30 51.94524   0.011       TRUE Employee 2.351669e-
Sickness  Net_A     30         1 44.00000   0.011       TRUE Employee 2.351669e-


> library(rpart)
> formula <- as.formula("EndObsW ~ Sex + TypeEmployee + TYPE_ARRET + Cause + Com
> maximal.tree <- rpart(formula, data = learning.sample, weights = KM.weight, me
```

Final tree estimator (CART) - ONLY THE AGE !



Predictions

```
> dim(validation.sample)
[1] 21261    17

> head(validation.sample)
   Sex TypeEmployee ContractNumber TYPE_ARRET   SurvDate BegIndDate EndIndDate
4    M          NCA        0006192    Maladie 2009-11-18 2009-12-18 2010-10-01 S
6    F          NCA        0024251    Maladie 2008-06-21 2008-07-21 2010-07-31 S
7    F          NCA        0037157    Maladie 2009-09-17 2009-10-17 2009-10-30 S
14   F          NCA        0099654    Maladie 2006-08-17 2006-09-16 2006-09-20 S
16   F          ENP        0119466    Maladie 2007-05-23 2007-06-22 2007-06-24 S
19   F          NCA        0154321    Maladie 2006-09-01 2006-10-01 2006-10-08 S

> predictions.validationSample <- predict(final.tree, newdata = validation.sampl

> summary(validation.sample$EndObsW)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0110  0.1533  0.4381  1.0290  1.1060 12.8600
> summary(predictions.validationSample) # notons que la moy. des previsions de d
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.7356  0.9682  1.2780  1.1820  1.5640  8.6830
```

```
> proba.nonCensure <- length(which(validation.sample$NonCensure == TRUE)) / nrow
> proba.nonCensure
[1] 0.9244626

> ## E[T|X] = E[T|delta = 1,X] P(delta=1) + E[T|delta = 0,X] P(delta=0)
> predictionsMoy.sinistresOuverts <- (mean(predictions.validationSample) - mean(
> provisionMoyenne <- predictionsMoy.sinistresOuverts * prestation.timeStep * nr
> provisionMoyenne
[1] 181022.9

> ## To be compared with:
> backtest.provisions.validationSample
[1] 179236.8

> ## Erreur de provision moyenne en pourcentage, backtesting:
> (abs(backtest.provisions.validationSample - provisionMoyenne) / max(c(provisio
[1] 0.9866959
```

And in the case of Chain Ladder ?

```
> triangle.cumule
```

	dev1	dev2	dev3	dev4	dev5	dev6	dev7	dev8	dev9	dev10
2006-01-01	44860	62511	72745	80289	85893	90337	93632	96355	98507	100076
2006-04-01	55982	76905	90518	101090	108863	115069	120081	123873	126825	129345
2006-07-01	49982	71709	84793	93874	100775	106524	110839	114411	117507	119784
2006-10-01	71692	101671	120151	133815	143029	149423	154704	158843	161888	164097
2007-01-01	63976	89524	104879	116125	123886	130064	135364	139655	143139	145725
2007-04-01	62908	87738	102469	113509	121848	128148	132965	136765	140138	143179
2007-07-01	57010	81126	96728	109027	118942	126480	132670	137549	141393	142766
2007-10-01	73432	102235	119236	131857	141478	149142	155474	160262	162374	NA
2008-01-01	69086	95648	111961	123578	131871	138414	143565	145966	NA	NA
2008-04-01	67486	93500	109196	120165	127846	133534	135821	NA	NA	NA
2008-07-01	62748	88588	102430	112289	119677	122728	NA	NA	NA	NA
2008-10-01	77569	107101	124141	136047	140492	NA	NA	NA	NA	NA
2009-01-01	66986	92879	107428	112450	NA	NA	NA	NA	NA	NA
2009-04-01	69909	96281	104723	NA	NA	NA	NA	NA	NA	NA
2009-07-01	58504	70612	NA	NA	NA	NA	NA	NA	NA	NA
2009-10-01	45583	NA	NA	NA	NA	NA	NA	NA	NA	NA

```
> CL.model <- chainladder(triangle.cumule)
> fact.dev <- sapply(CL.model$Models, coef) # a comparer avec 'fact.dev.CL' calc
> fact.dev
       x        x        x        x        x        x        x        x        x
1.384294 1.163524 1.102059 1.067753 1.049660 1.037865 1.029157 1.022532 1.016758
```

```
> rectangle.cumule
            dev
origin        dev1      dev2      dev3      dev4      dev5      dev6      dev7
  2006-01-01 44860  62511.00  72745.00  80289.00  85893.00  90337.00  93632.00
  2006-04-01 55982  76905.00  90518.00 101090.00 108863.00 115069.00 120081.00 1
  2006-07-01 49982  71709.00  84793.00  93874.00 100775.00 106524.00 110839.00 1
  2006-10-01 71692 101671.00 120151.00 133815.00 143029.00 149423.00 154704.00 1
  2007-01-01 63976  89524.00 104879.00 116125.00 123886.00 130064.00 135364.00 1
  2007-04-01 62908  87738.00 102469.00 113509.00 121848.00 128148.00 132965.00 1
  2007-07-01 57010  81126.00  96728.00 109027.00 118942.00 126480.00 132670.00 1
  2007-10-01 73432 102235.00 119236.00 131857.00 141478.00 149142.00 155474.00 1
  2008-01-01 69086  95648.00 111961.00 123578.00 131871.00 138414.00 143565.00 1
  2008-04-01 67486  93500.00 109196.00 120165.00 127846.00 133534.00 135821.00 1
  2008-07-01 62748  88588.00 102430.00 112289.00 119677.00 122728.00 127375.09 1
  2008-10-01 77569 107101.00 124141.00 136047.00 140492.00 147468.81 153052.71 1
  2009-01-01 66986  92879.00 107428.00 112450.00 120068.87 126031.47 130803.65 1
  2009-04-01 69909  96281.00 104723.00 115410.90 123230.38 129349.99 134247.82 1
  2009-07-01 58504  70612.00  82158.73  90543.75  96678.40 101479.43 105321.95 1
  2009-10-01 45583  63100.28  73418.67  80911.69  86393.73  90684.03  94117.78
```

```
  > cbind(Provision.parExercice)
      Provision.parExercice
 [1,]          0.000000e+00
 [2,]          0.000000e+00
 [3,]         -7.275958e-11
 [4,]         -8.731149e-11
 [5,]          2.209210e+01
 [6,]          7.691398e+02
 [7,]          2.403482e+03
 [8,]          5.500488e+03
 [9,]          8.345020e+03
[10,]          1.195160e+04
[11,]          1.585549e+04
[12,]          2.602862e+04
[13,]          2.986374e+04
[14,]          4.133798e+04
[15,]          4.397777e+04
[16,]          5.681669e+04
> (Provision.globale <- sum(Provision.parExercice))
[1] 242872.1
```

```
> ## Erreur de calcul de provision moyenne par Chain Ladder, backtesting:
> (abs(backtest.provisions.validationSample - Provision.globale) / max(c(Provisi
[1] 26.20117
```

Final remarks

+ Particularly adapted to long-development claims ;
+ Discriminating power of covariates.
+ Simple and easy-to-understand final estimator.

+ Consistent procedure and theoretical guarantees.
+ Extensions by working on the loss function.

- Instability : need to gain robustness (random forests, ...).

Still to do : use bootstrap resampling technique to get some confidence interval around the estimation.