

Claims reserving with R: ChainLadder-0.1.5-2 Package Vignette

Markus Gesmann*

January 14, 2012

Abstract

The `ChainLadder` package provides various statistical methods which are typically used for the calculation of outstanding claims reserves in general insurance.

The package has implementations of the Mack-, Munich-, Bootstrap, and multi-variate chain-ladder methods, as well as the loss development factor curve fitting methods of Dave Clark and generalised linear model based reserving models.

*markus.gesmann@gmail.com

Contents

1	Introduction	4
1.1	Claims reserving in insurance	4
2	The ChainLadder package	4
2.1	Motivation	4
2.2	Brief package overview	5
2.3	Installation	6
3	Using the ChainLadder package	7
3.1	Working with triangles	7
3.1.1	Plotting triangles	7
3.1.2	Transforming triangles between cumulative and incremental representation	9
3.1.3	Importing triangles from external data sources	10
3.1.4	Coping and pasting from MS Excel	12
3.2	Chain-ladder methods	13
3.2.1	Mack chain-ladder	13
3.2.2	Bootstrap chain-ladder	15
3.2.3	Munich chain-ladder	18
3.3	Multivariate chain-ladder	20
3.4	Clark's methods	20
3.4.1	Clark's Cap Cod method	20
3.4.2	Clark's LDF method	20
3.5	Generalised linear model methods	20
4	Using ChainLadder with RExcel and SWord	20
5	Further resources	21
5.1	Other insurance related R packages	21
5.2	Presentations	22
5.3	Further reading	22

6 Training and consultancy	22
References	25
Thanks	26

1 Introduction

1.1 Claims reserving in insurance

Unlike other industries the insurance industry does not sell products as such, but promises. An insurance policy is a promise by the insurer to the policyholder to pay for future claims for an upfront received premium.

As a result insurers don't know the upfront cost of their service, but rely on data analysis and judgement to derive a sustainable price for their offering. In General Insurance (or Non-Life Insurance) most policies run for a period of 12 months, e.g. motor, property and casualty insurance. However, the claims payment process can take years or even decades.

In particular claims arising from casualty insurance can take a long time to settle, as claims can take years to materialise. A complex and costly example are the claims from asbestos liabilities. A research report by a Working Party of the Institute of Actuaries has estimated that the undiscounted cost of UK mesothelioma-related claims to the UK Insurance Market for the period 2009 to 2050 could be around £10bn [GBB⁺09]. The cost for Asbestos related claims in the US for the worldwide insurance industry was estimate to be around \$120bn in 2002 [Mic02].

Thus, it should come to no surprise that the biggest item on the liability side of an insurer's balance sheet is often the provision or reserves for future claims payments. Those reserves can be broken down in case reserves (or out-standings claims), which are losses already reported to the insurance company and incurred but not reported (IBNR) claims.

Over the years several methods have been developed to estimate reserves for insurance claims, see [Sch11], [PR02] for an overview. Changes in the regulatory requirements, e.g. Solvency II¹ in Europe, have fostered further research into this topic, with a focus on stochastic and statistical techniques.

2 The ChainLadder package

2.1 Motivation

The ChainLadder [GMZ12] package provides various statistical methods which are typically used for the calculation of outstanding claims reserves in general insurance. The package started out of presentations given by Markus Gesmann at the Stochastic Reserving Seminar at the Institute of Actuaries in 2007 and 2008, followed by talks at Casualty Actuarial Society (CAS) meetings joined by Dan Murphy in 2008 and Wayne Zhang in 2010.

Implementing reserving methods in R has several advantages, as it provides:

¹See http://ec.europa.eu/internal_market/insurance/solvency/index_en.htm

- Rich language for statistical modelling and data manipulations allowing fast prototyping
- Very active user base, which publishes many extension
- Many interfaces to data bases and other applications, such as MS Excel
- Established framework for documentation and testing
- Works well with version control tools
- Code is written in plain text files, allowing effective knowledge transfer
- Easy to collaborate over the internet
- Built in functions to create reproducible research reports²
- In combination with other tools such as \LaTeX and Sweave easy to set up automated reporting
- Academic research is often first implemented in R

2.2 Brief package overview

This vignette will give the reader a brief overview of the functionality of the ChainLadder package. The functions are discussed and explained in more detail in the respective help files and examples. The help files contain the references to the research papers on which the methods are based.

The ChainLadder package has implementations of the Mack-, Munich- and Bootstrap chain-ladder methods [Mac93], [Mac99], [QM04], [EV99]. Since version 0.1.3-3 it provides general multivariate chain ladder models by Wayne Zhang [Zha10].

Version 0.1.4-0 introduced new functions on loss development factor (LDF) fitting and Cape Cod by Daniel Murphy following a paper by David Clark [Cla03]. Version 0.1.5-0 has added loss reserving models within the generalized linear model framework following a paper by England and Verrall [EV99] implemented by Wayne Zhang.

The package offers also some utility functions to convert quickly tables into triangles, triangles into tables, cumulative into incremental and incremental into cumulative triangles.

A set of demos is shipped with the packages and the list of demos names is available via:

```
R> demo(package="ChainLadder")
```

and can be executed via

²For an example see the project: Formatted Actuarial Vignettes in R, <http://www.favir.net/>

```
R> library(ChainLadder)
R> demo("demo name")
```

Further, the ChainLadder package comes with example files, which demonstrates how to the ChainLadder functions can be embedded in Excel and Word using the [statconn](#) interface[BN07].

For more information and examples see the project web site: <http://code.google.com/p/chainladder/>

2.3 Installation

We can install ChainLadder in the usual way from CRAN, e.g.:

```
R> install.packages('ChainLadder')
```

The installation was successful if the command `library(ChainLadder)` gives you the following message:

```
R> library(ChainLadder)
```

```
ChainLadder version 0.1.5-2 by:
Markus Gesmann <markus.gesmann@gmail.com>
Wayne Zhang <actuary_zhang@hotmail.com>
Daniel Murphy <danielmarkmurphy@gmail.com>
```

```
Type library(help='ChainLadder') or ?ChainLadder
to see overall documentation.
```

```
Type demo(ChainLadder) to get an idea of the functionality of this package.
```

```
See demo(package='ChainLadder') for a list of more demos.
```

```
Feel free to send us an email if you would like to keep informed of
new versions or if you have any feedback, ideas, suggestions or would
like to collaborate.
```

```
More information is available on the ChainLadder project web-site:
http://code.google.com/p/chainladder/
```

```
To suppress this message use the statement:
suppressPackageStartupMessages(library(ChainLadder))
```

3 Using the ChainLadder package

3.1 Working with triangles

Historical insurance data is often presented in form of a triangle structure, showing the development of claims over time for each origin period. An origin period could be the year the policy was sold, or the accident year. Most reserving methods of the ChainLadder package expect triangles as input data sets with development periods along the columns and the origin period in rows. The package comes with several example triangles. The following R command will list them all:

```
R> require(ChainLadder)
R> data(package="ChainLadder")
```

Let's look at one example triangle more closely. The following triangle shows data from the Reinsurance Association of America (RAA):

```
R> ## Sample triangle
R> RAA
```

	dev									
origin	1	2	3	4	5	6	7	8	9	10
1981	5012	8269	10907	11805	13539	16181	18009	18608	18662	18834
1982	106	4285	5396	10666	13782	15599	15496	16169	16704	NA
1983	3410	8992	13873	16141	18735	22214	22863	23466	NA	NA
1984	5655	11555	15766	21266	23425	26083	27067	NA	NA	NA
1985	1092	9565	15836	22169	25955	26180	NA	NA	NA	NA
1986	1513	6445	11702	12935	15852	NA	NA	NA	NA	NA
1987	557	4020	10946	12314	NA	NA	NA	NA	NA	NA
1988	1351	6947	13112	NA	NA	NA	NA	NA	NA	NA
1989	3133	5395	NA	NA	NA	NA	NA	NA	NA	NA
1990	2063	NA	NA	NA	NA	NA	NA	NA	NA	NA

The objective of a reserving exercise is to forecast the future claims development in the bottom right corner of the triangle and potential further developments. Eventually all claims for a given origin period will be settled, but it is not always obvious to judge how many years or even decades it will take. We speak of long and short tail business depending on the time it takes to finalise all claims reporting.

3.1.1 Plotting triangles

The first thing you often want to do is to plot the data to get an overview. For data set of class `triangle` ChainLadder provides default plotting methods to give a graphical overview of the data:

```
R> plot(RAA)
```

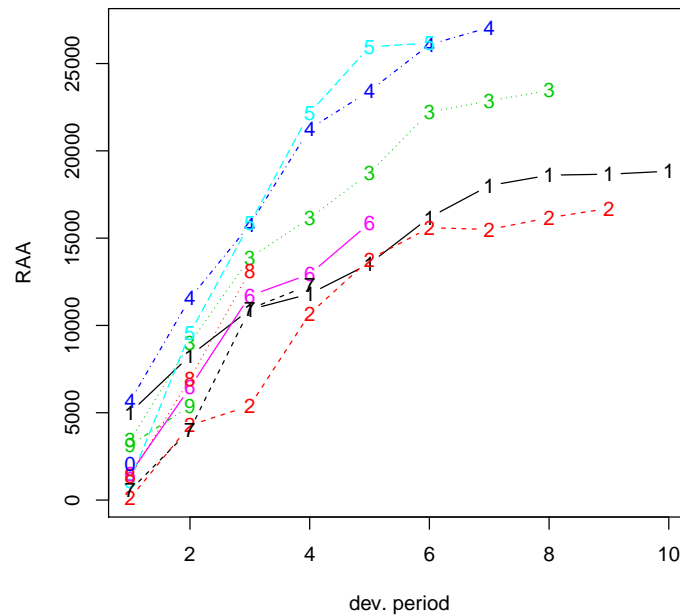


Figure 1: Claims development chart of the RAA triangle, with one line per origin period. Output of `plot(RAA)`

Setting the argument `lattice=TRUE` will produce individual plots for each origin period³, see Figure 2.

```
R> plot(RAA, lattice=TRUE)
```

You will notice from the plots in Figures 1 and 2 that the triangle RAA presents claims developments for the origin years 1981 to 1990 in a cumulative form. For more information on the triangle plotting functions see the help pages of `plot.triangle`, e.g. via

```
R> ?plot.triangle
```

³ChainLadder uses the `lattice` package

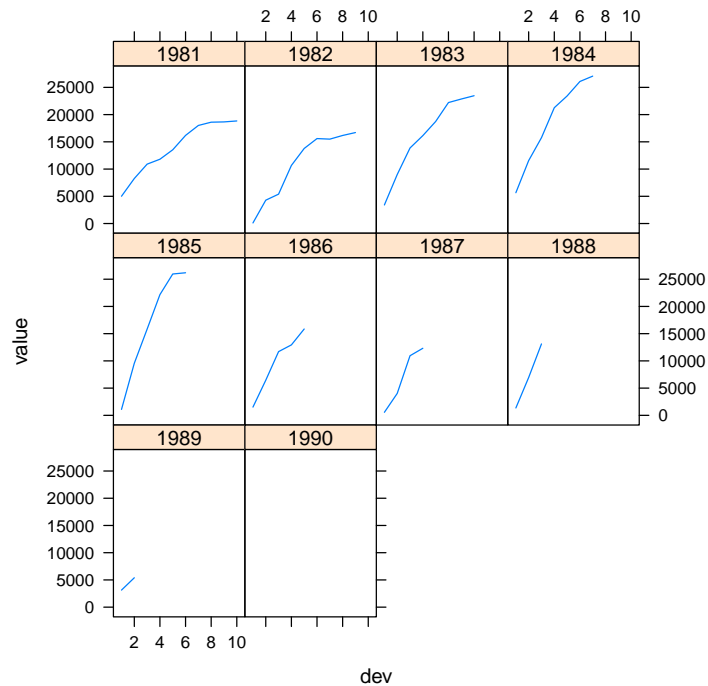


Figure 2: Claims development chart of the RAA triangle, with individual panels for each origin period. Output of `plot(RAA, lattice=TRUE)`

3.1.2 Transforming triangles between cumulative and incremental representation

The ChainLadder packages comes with two helper functions, `cum2incr` and `incr2cum` to transform cumulative triangles into incremental triangles and vis versa:

```
R> raa.inc <- cum2incr(RAA)
R> ## Show first origin period and its incremental development
R> raa.inc[1,]
```

```
      1      2      3      4      5      6      7      8      9     10
5012 3257 2638  898 1734 2642 1828  599  54  172
```

```
R> raa.cum <- incr2cum(raa.inc)
R> ## Show first origin period and its cumulative development
R> raa.cum[1,]
```

1	2	3	4	5	6	7	8	9	10
5012	8269	10907	11805	13539	16181	18009	18608	18662	18834

3.1.3 Importing triangles from external data sources

In most cases you want to analyse your own data, usually stored in data bases. R makes it easy to access data using SQL statements, e.g. via an ODBC connection⁴ and the `ChainLadder` packages includes a demo to showcase how data can be imported from a MS Access data base, see:

```
R> demo(DatabaseExamples)
```

For more details see [Tea11].

In this section we use data stored in a CSV-file⁵ to demonstrate some typical operations you will want to carry out with data stored in data bases. In most cases your triangles will be stored in tables and not in a classical triangle shape. The `ChainLadder` package contains a CSV-file with some sample data in a long table format. We can read the data into R's memory with the `read.csv` command and look at the first couple of rows and summarise it:

```
R> filename <- file.path(system.file("Database",
+                                   package="ChainLadder"),
+                         "TestData.csv")
R> myData <- read.csv(filename)
R> head(myData)
```

	origin	dev	value	lob
1	1977	1	153638	ABC
2	1978	1	178536	ABC
3	1979	1	210172	ABC
4	1980	1	211448	ABC
5	1981	1	219810	ABC
6	1982	1	205654	ABC

```
R> summary(myData)
```

	origin	dev	value		lob
Min.	: 1	Min. : 1.00	Min. : -17657	AutoLiab	:105
1st Qu.:	3	1st Qu.: 2.00	1st Qu.: 10324	GenerallLiab	:105
Median :	6	Median : 4.00	Median : 72468	M3IR5	:105

⁴See the `RODBC` package

⁵Please ensure that your CSV-file is free from formatting, e.g. characters to separate units of thousands, as columns with those kind of formatting would be read as characters.

Mean	: 642	Mean	: 4.61	Mean	: 176632	ABC	: 66
3rd Qu.:	1979	3rd Qu.:	7.00	3rd Qu.:	197716	CommercialAutoPaid:	55
Max.	:1991	Max.	:14.00	Max.	:3258646	GenIns	: 55
						(Other)	:210

Let us focus on one subset of the data set. We select the RAA data again:

```
R> raa <- subset(myData, lob %in% "RAA")
R> head(raa)
```

	origin	dev	value	lob
67	1981	1	5012	RAA
68	1982	1	106	RAA
69	1983	1	3410	RAA
70	1984	1	5655	RAA
71	1985	1	1092	RAA
72	1986	1	1513	RAA

To transform the long table of the RAA data into a triangle we use the function `as.triangle`. The arguments we have to specify are the column names of the origin and development period and further the column which contains the values:

```
R> raa.tri <- as.triangle(raa,
+                          origin="origin",
+                          dev="dev",
+                          value="value")
R> raa.tri
```

	dev										
origin	1	2	3	4	5	6	7	8	9	10	
1981	5012	3257	2638	898	1734	2642	1828	599	54	172	
1982	106	4179	1111	5270	3116	1817	-103	673	535	NA	
1983	3410	5582	4881	2268	2594	3479	649	603	NA	NA	
1984	5655	5900	4211	5500	2159	2658	984	NA	NA	NA	
1985	1092	8473	6271	6333	3786	225	NA	NA	NA	NA	
1986	1513	4932	5257	1233	2917	NA	NA	NA	NA	NA	
1987	557	3463	6926	1368	NA	NA	NA	NA	NA	NA	
1988	1351	5596	6165	NA	NA	NA	NA	NA	NA	NA	
1989	3133	2262	NA	NA	NA	NA	NA	NA	NA	NA	
1990	2063	NA	NA	NA	NA	NA	NA	NA	NA	NA	

We note that the data has been stored as an incremental data set. As mentioned above, we could now use the function `incr2cum` to transform the triangle into a cumulative format.

We can transform a triangle back into a data frame structure:

```
R> raa.df <- as.data.frame(raa.tri, na.rm=TRUE)
R> head(raa.df)
```

	origin	dev	value
1981-1	1981	1	5012
1982-1	1982	1	106
1983-1	1983	1	3410
1984-1	1984	1	5655
1985-1	1985	1	1092
1986-1	1986	1	1513

This is particular helpful when you would like to store your results back into data base. Figure 3 gives you an idea of a potential data flow between R and data bases.

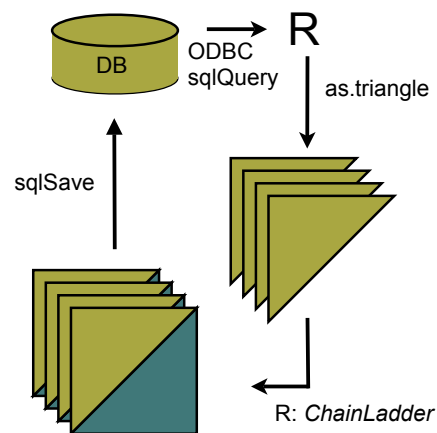


Figure 3: Flow chart of data between R and data bases.

3.1.4 Coping and pasting from MS Excel

Small data sets in Excel can be transferred to R backwards and forwards with via the clipboard under MS Windows.

Copying from Excel to R Select a data set in Excel and copy it into the clipboard, then go to R and type:

```
R> x <- read.table(file="clipboard", sep="\t", na.strings="")
```

Copying from R to Excel Suppose you would like to copy the RAA triangle into Excel, then the following statement would copy the data into the clipboard:

```
R> write.table(RAA, file="clipboard", sep="\t", na="")
```

Now you can paste the content into Excel. Please note that you can not copy lists structures from R to Excel.

3.2 Chain-ladder methods

```
R> demo(ChainLadder)
```

3.2.1 Mack chain-ladder

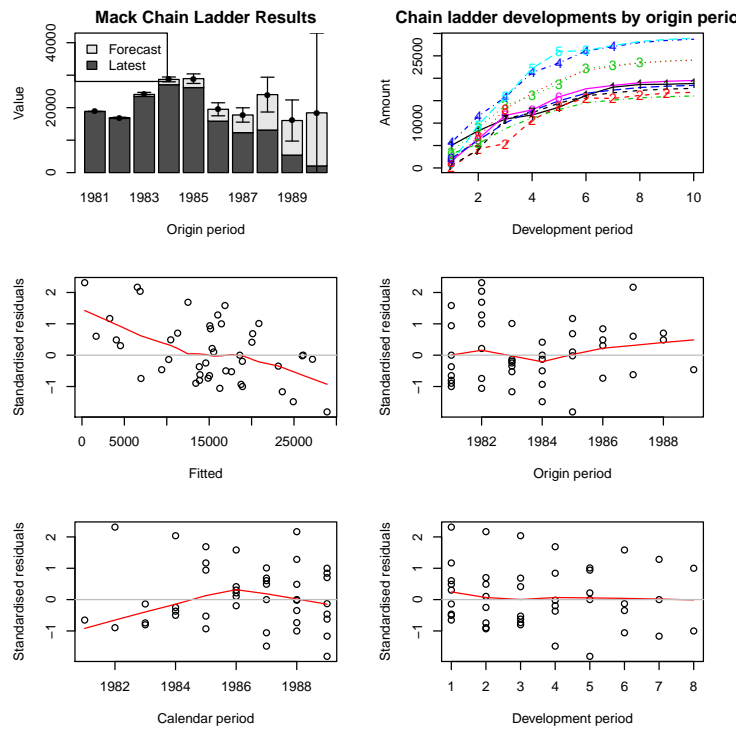
```
R> mack <- MackChainLadder(RAA, est.sigma="Mack")
R> mack
```

```
MackChainLadder(Triangle = RAA, est.sigma = "Mack")
```

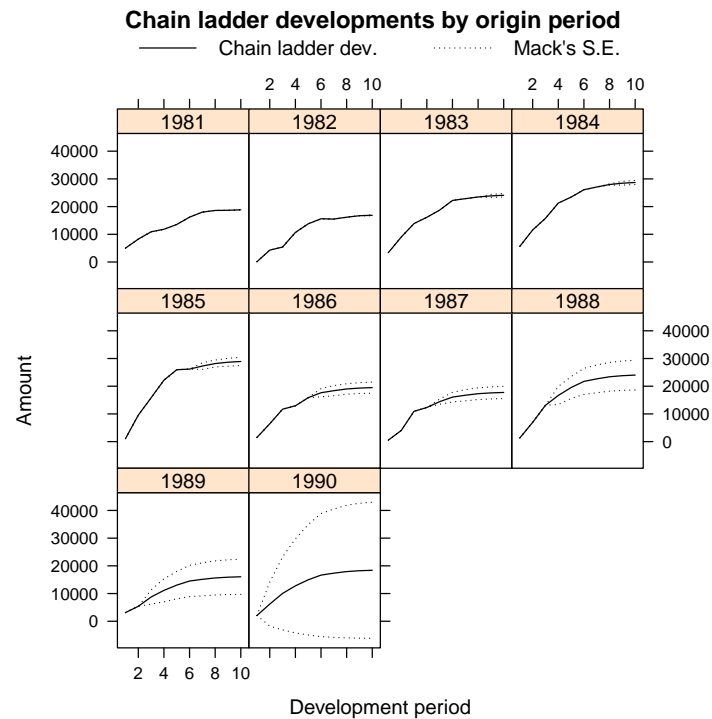
	Latest	Dev.To.Date	Ultimate	IBNR	Mack.S.E	CV(IBNR)
1981	18,834	1.000	18,834	0	0	NaN
1982	16,704	0.991	16,858	154	206	1.339
1983	23,466	0.974	24,083	617	623	1.010
1984	27,067	0.943	28,703	1,636	747	0.457
1985	26,180	0.905	28,927	2,747	1,469	0.535
1986	15,852	0.813	19,501	3,649	2,002	0.549
1987	12,314	0.694	17,749	5,435	2,209	0.406
1988	13,112	0.546	24,019	10,907	5,358	0.491
1989	5,395	0.336	16,045	10,650	6,333	0.595
1990	2,063	0.112	18,402	16,339	24,566	1.503

Totals	
Latest:	160,987.00
Dev:	0.76
Ultimate:	213,122.23
IBNR:	52,135.23
Mack S.E.:	26,909.01
CV(IBNR):	0.52

```
R> plot(mack)
```



```
R> plot(mack, lattice=TRUE)
```



3.2.2 Bootstrap chain-ladder

R> # See also the example in section 8 of England & Verrall (2002) on page 55.

R>

R> B <- BootChainLadder(RAA, R=999, process.distr="gamma")

R> B

`BootChainLadder(Triangle = RAA, R = 999, process.distr = "gamma")`

	Latest	Mean	Ultimate	Mean	IBNR	SD	IBNR	IBNR	75%	IBNR	95%
1981	18,834		18,834		0		0		0		0
1982	16,704		16,865		161		671		191		1,416
1983	23,466		24,094		628		1,246		1,030		2,963
1984	27,067		28,825		1,758		1,949		2,758		5,439
1985	26,180		29,064		2,884		2,409		4,319		7,601
1986	15,852		19,636		3,784		2,469		5,166		8,102
1987	12,314		17,981		5,667		3,162		7,466		11,559
1988	13,112		24,255		11,143		5,060		13,988		20,197
1989	5,395		16,368		10,973		6,274		14,483		22,697
1990	2,063		19,713		17,650		13,555		25,363		43,594

```

Totals
Latest:      160,987
Mean Ultimate: 215,636
Mean IBNR:   54,649
SD IBNR:     18,963
Total IBNR 75%: 65,835
Total IBNR 95%: 87,421

```

```

R> plot(B)
R> # Compare to MackChainLadder
R> MackChainLadder(RAA)

```

```

MackChainLadder(Triangle = RAA)

```

	Latest	Dev.To.Date	Ultimate	IBNR	Mack.S.E	CV(IBNR)
1981	18,834	1.000	18,834	0	0	NaN
1982	16,704	0.991	16,858	154	143	0.928
1983	23,466	0.974	24,083	617	592	0.959
1984	27,067	0.943	28,703	1,636	713	0.436
1985	26,180	0.905	28,927	2,747	1,452	0.529
1986	15,852	0.813	19,501	3,649	1,995	0.547
1987	12,314	0.694	17,749	5,435	2,204	0.405
1988	13,112	0.546	24,019	10,907	5,354	0.491
1989	5,395	0.336	16,045	10,650	6,332	0.595
1990	2,063	0.112	18,402	16,339	24,566	1.503

```

Totals
Latest:      160,987.00
Dev:         0.76
Ultimate:    213,122.23
IBNR:        52,135.23
Mack S.E.:   26,880.74
CV(IBNR):    0.52

```

```

R> quantile(B, c(0.75,0.95,0.99, 0.995))

```

```

$ByOrigin
IBNR 75% IBNR 95% IBNR 99% IBNR 99.5%
1981      0.0      0      0      0
1982    190.9    1416    2663    3284
1983   1030.5   2963    4617    5241
1984   2758.2   5439    7767    9004
1985   4319.1   7601    9666   9981
1986   5165.6   8102   11686  12326
1987   7465.9  11559   15167  15319

```


1988	13987.6	20197	26489	27254
1989	14482.7	22697	29008	31582
1990	25362.6	43594	56806	60296

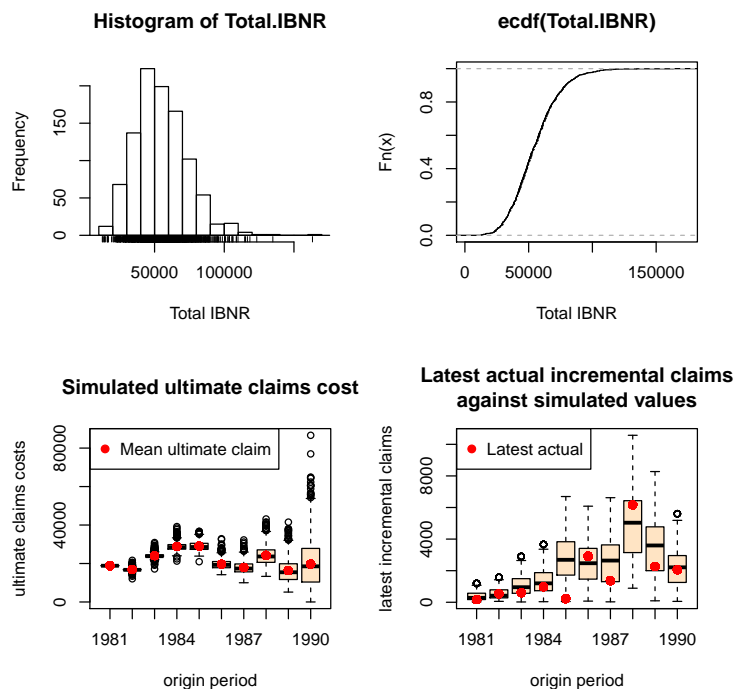
\$Totals

	Totals
IBNR 75%:	65835
IBNR 95%:	87421
IBNR 99%:	106253
IBNR 99.5%:	115272

```
R> # fit a distribution to the IBNR
R> library(MASS)
R> plot(ecdf(B$IBNR.Totals))
R> # fit a log-normal distribution
R> fit <- fitdistr(B$IBNR.Totals[B$IBNR.Totals>0], "lognormal")
R> fit
```

meanlog	sdlog
10.846301	0.363454
(0.011499)	(0.008131)

```
R> curve(plnorm(x,fit$estimate["meanlog"], fit$estimate["sdlog"]), col="red", add=TRUE)
R>
```



3.2.3 Munich chain-ladder

R> MCLpaid

	dev						
origin	1	2	3	4	5	6	7
1	576	1804	1970	2024	2074	2102	2131
2	866	1948	2162	2232	2284	2348	NA
3	1412	3758	4252	4416	4494	NA	NA
4	2286	5292	5724	5850	NA	NA	NA
5	1868	3778	4648	NA	NA	NA	NA
6	1442	4010	NA	NA	NA	NA	NA
7	2044	NA	NA	NA	NA	NA	NA

R> MCLincurred

	dev						
origin	1	2	3	4	5	6	7
1	978	2104	2134	2144	2174	2182	2174
2	1844	2552	2466	2480	2508	2454	NA

3	2904	4354	4698	4600	4644	NA	NA
4	3502	5958	6070	6142	NA	NA	NA
5	2812	4882	4852	NA	NA	NA	NA
6	2642	4406	NA	NA	NA	NA	NA
7	5022	NA	NA	NA	NA	NA	NA

```
R> op <- par(mfrow=c(1,2))
R> plot(MCLpaid)
R> plot(MCLincurred)
R> par(op)
R> # Following the example in Quarg's (2004) paper:
R> MCL <- MunichChainLadder(MCLpaid, MCLincurred, est.sigmaP=0.1, est.sigmaI=0.1)
R> MCL
```

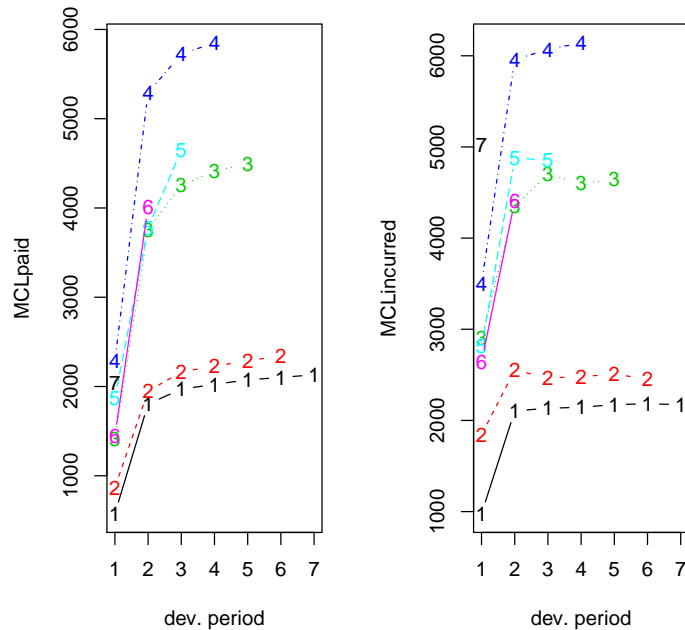
```
MunichChainLadder(Paid = MCLpaid, Incurred = MCLincurred, est.sigmaP = 0.1,
  est.sigmaI = 0.1)
```

	Latest Paid	Latest Incurred	Latest P/I Ratio	Ult. Paid	Ult. Incurred
1	2,131	2,174	0.980	2,131	2,174
2	2,348	2,454	0.957	2,383	2,444
3	4,494	4,644	0.968	4,597	4,629
4	5,850	6,142	0.952	6,119	6,176
5	4,648	4,852	0.958	4,937	4,950
6	4,010	4,406	0.910	4,656	4,665
7	2,044	5,022	0.407	7,549	7,650

	Ult. P/I Ratio
1	0.980
2	0.975
3	0.993
4	0.991
5	0.997
6	0.998
7	0.987

Totals	Paid	Incurred	P/I Ratio
Latest:	25,525	29,694	0.86
Ultimate:	32,371	32,688	0.99

```
R> plot(MCL)
```



3.3 Multivariate chain-ladder

3.4 Clark's methods

3.4.1 Clark's Cap Cod method

3.4.2 Clark's LDF method

3.5 Generalised linear model methods

4 Using ChainLadder with RExcel and SWord

The spreadsheet is located in the Excel folder of the package. The R command

```
R> system.file("Excel", package="ChainLadder")
```

will tell you the exact path to the directory. To use the spreadsheet you will need the RExcel-Add-in [BN07]. The package also provides an example SWord file, demonstrating how the the functions of the package can be integrated into a MS Word file via SWord [BN07]. Again you find the Word file via the command:

```
R> system.file("SWord", package="ChainLadder")
```

The package comes with several demos to provide you with an overview of the package functionality, see

```
R> demo(package="ChainLadder")
```

5 Further resources

Other useful documents and resources to get started with R in the context of actuarial work:

- Introduction to R for Actuaries [DS06].
- An Actuarial Toolkit [MSH⁺06].
- Actuar package vignettes: <http://cran.r-project.org/web/packages/actuar/index.html>
- Mailing list [R-SIG-insurance](#)⁶: Special Interest Group on using R in actuarial science and insurance

5.1 Other insurance related R packages

Below is a list of further R packages in the context of insurance. The list is by no means complete, and the CRAN Task Views '[Emperical Finance](#)' and '[Probability Distributions](#)' will provide links to additional resources. Please feel free to contact [us](#) with items to be added to the list.

- `cp1m`: Monte Carlo EM algorithms and Bayesian methods for fitting Tweedie compound Poisson linear models [Zha11].
- `lossDev`: A Bayesian time series loss development model. Features include skewed-t distribution with time-varying scale parameter, Reversible Jump MCMC for determining the functional form of the consumption path, and a structural break in this path [LS11].
- `favir`: Formatted Actuarial Vignettes in R. FAViR lowers the learning curve of the R environment. It is a series of peer-reviewed Sweave papers that use a consistent style [Esc11].
- `actuar`: Loss distributions modelling, risk theory (including ruin theory), simulation of compound hierarchical models and credibility theory [DGP08].

⁶<https://stat.ethz.ch/mailman/listinfo/r-sig-insurance>

- `fitdistrplus`: Help to fit of a parametric distribution to non-censored or censored data [DMPDD10].
- `mondate`: R packackge to keep track of dates in terms of months [Mur11].
- `lifecontingencies`: Package to perform actuarial evaluation of life contingencies [Spe11].

5.2 Presentations

Over the years the contributors of the ChainLadder package have given numerous presentations and most of those are still available online:

- [Bayesian Hierarchical Models in Property-Casualty Insurance](#), Wayne Zhang, 2011
- [ChainLadder at the Predictive Modelling Seminar, Institute of Actuaries, November 2010](#), Markus Gesmann, 2011
- [Reserve variability calculations](#), CAS spring meeting, San Diego, Jimmy Curcio Jr., Markus Gesmann and Wayne Zhang, 2010
- [The ChainLadder package, working with databases and MS Office interfaces, presentation at the "R you ready?" workshop](#), Institute of Actuaries, Markus Gesmann, 2009
- [The ChainLadder package](#), London R user group meeting, Markus Gesmann, 2009
- [Introduction to R, Loss Reserving with R](#), Stochastic Reserving and Modelling Seminar, Institute of Actuaries, Markus Gesmann, 2008
- [Loss Reserving with R](#), CAS meeting, Vincent Goulet, Markus Gesmann and Daniel Murphy, 2008
- [The ChainLadder package](#) R-user conference Dortmund, Markus Gesmann, 2008

5.3 Further reading

Other papers and presentation which cited ChainLadder : [Orr07], [Nic09], [Zha10], [MNNV10], [Sch10], [MNV10], [Esc11], [Spe11]

6 Training and consultancy

Please contact [us](#) if you would like to discuss tailored training or consultancy.

References

- [BN07] Thomas Baier and Erich Neuwirth. Excel :: Com :: R. *Computational Statistics*, 22(1), April 2007. Physica Verlag.
- [Cla03] David R. Clark. *LDF Curve-Fitting and Stochastic Reserving: A Maximum Likelihood Approach*. Casualty Actuarial Society, 2003. CAS Fall Forum.
- [DGP08] C Dutang, V. Goulet, and M. Pigeon. actuar: An R package for actuarial science. *Journal of Statistical Software*, 25(7), 2008.
- [DMPDD10] Marie Laure Delignette-Muller, Regis Pouillot, Jean-Baptiste Denis, and Christophe Dutang. *fitdistrplus: help to fit of a parametric distribution to non-censored or censored data*, 2010. R package version 0.1-3.
- [DS06] Nigel De Silva. An introduction to r: Examples for actuaries. <http://toolkit.pbwiki.com/RToolkit>, 2006.
- [Esc11] Benedict Escoto. *favir: Formatted Actuarial Vignettes in R*, 0.5-1 edition, January 2011.
- [EV99] Peter England and Richard Verrall. Analytic and bootstrap estimates of prediction errors in claims reserving. *Mathematics and Economics*, Vol. 25:281 – 293, 1999.
- [GBB⁺09] Brian Gravelsons, Matthew Ball, Dan Beard, Robert Brooks, Naomi Couchman, Brian Gravelsons, Charlie Kefford, Darren Michaels, Patrick Nolan, Gregory Overton, Stephen Robertson-Dunn, Emiliano Ruffini, Graham Sandhouse, Jerome Schilling, Dan Sykes, Peter Taylor, Andy Whiting, Matthew Wilde, and John Wilson. B12: Uk asbestos working party update 2009. <http://www.actuaries.org.uk/research-and-resources/documents/b12-uk-asbestos-working-party-update-2009-5mb>, October 2009. Presented at the General Insurance Convention.
- [GMZ12] Markus Gesmann, Dan Murphy, and Wayne Zhang. *ChainLadder: Mack-, Bootstrap and Munich-chain-ladder methods for insurance claims reserving*, 2012. R package version 0.1.5-2.
- [LS11] Christopher W. Laws and Frank A. Schmid. *lossDev: Robust Loss Development Using MCMC*, 2011. R package version 3.0.0-1.
- [Mac93] Thomas Mack. Distribution-free calculation of the standard error of chain ladder reserve estimates. *Astin Bulletin*, Vol. 23:213 – 25, 1993.
- [Mac99] Thomas Mack. The standard error of chain ladder reserve estimates: Recursive calculation and inclusion of a tail factor. *Astin Bulletin*, Vol. 29(2):361 – 266, 1999.

- [Mic02] Darren Michaels. APH: how the love carnal and silicone implants nearly destroyed Lloyd's (slides). <http://www.actuaries.org.uk/research-and-resources/documents/aph-how-love-carnal-and-silicone-implants-nearly-destroyed-lloyds-s>, December 2002. Presented at the Younger Members' Convention.
- [MNNV10] Maria Dolores Martinez Miranda, Bent Nielsen, Jens Perch Nielsen, and Richard Verrall. *Cash flow simulation for a model of outstanding liabilities based on claim amounts and claim numbers*. CASS, September 2010.
- [MNV10] Maria Dolores Martinez Miranda, Jens Perch Nielsen, and Richard Verrall. *Double Chain Ladder*. ASTIN, Colloquia Madrid edition, 2010.
- [MSH⁺06] Trevor Maynard, Nigel De Silva, Richard Holloway, Markus Gesmann, Sie Lau, and John Harnett. An actuarial toolkit. introducing The Toolkit Manifesto. <http://www.actuaries.org.uk/sites/all/files/documents/pdf/actuarial-toolkit.pdf>, 2006. General Insurance Convention.
- [Mur11] Dan Murphy. *mondate: Keep track of dates in terms of months*, 2011. R package version 0.9.8.24.
- [Nic09] Luke Nichols. *Multimodel Inference for Reserving*. Australian Prudential Regulation Authority (APRA), December 2009.
- [Orr07] James Orr. *A Simple Multi-State Reserving Model*. ASTIN, Colloquia Orlando edition, 2007.
- [PR02] P.D.England and R.J.Verrall. Stochastic claims reserving in general insurance. *British Actuarial Journal*, 8:443–544, 2002.
- [QM04] Gerhard Quarg and Thomas Mack. Munich chain ladder. Munich Re Group, 2004.
- [Sch10] Ernesto Schirmacher. Reserve variability calculations, chain ladder, R, and Excel. <http://www.casact.org/affiliates/cane/0910/schirmacher.pdf>, September 2010. Presentation at the Casualty Actuaries of New England (CANE) meeting.
- [Sch11] Klaus D. Schmidt. A bibliography on loss reserving. <http://www.math.tu-dresden.de/sto/schmidt/dsvm/reserve.pdf>, 2011.
- [Spe11] Giorgio Alfredo Spedicato. *Introduction to lifecontingencies Package*. StatisticalAdvisor Inc, 0.0.4 edition, November 2011.
- [Tea11] R Development Core Team. *R Data Import/Export*. R Foundation for Statistical Computing, 2011. ISBN 3-900051-10-0.

- [Zha10] Y. Zhang. A general multivariate chain ladder model. *Insurance: Mathematics and Economics*, 46:588 – 599, 2010.
- [Zha11] Wayne Zhang. *cplm: Tweedie compound Poisson linear models*, 2011. R package version 0.4-1.

Thanks

Many thanks to all who provided ideas, suggestions, corrections and bug reports:

- Nigel de Silva for all the ideas on how to use arrays effiecently
- Florian Leitenstorfer for a bug report on MackChainLadder
- Beat Huggler for comments on MunichChainLadder
- Daniel Murphy for comments on MackChainLadder
- Mark Hoffmann for a bug report on MackChainLadder
- Christophe Dutang for ideas and code on utility functions to deal with triangles
- Stefan Pohl for comments on tail factors with MunichChainLadder
- Ben Escoto for providing a patch to a bug on returning latest incomplete triangle positions
- Przemyslaw Sloma for reporting a bug in MackChainLadder
- Ernesto Schirmacher for reporting a bug in residuals.MackChainLadder