Using R in insurance

Markus Gesmann*,

Lloyd's, Market Analysis

GIRO Convention, Vienna, 26-29 September 2006

toolkit\ RToolkit

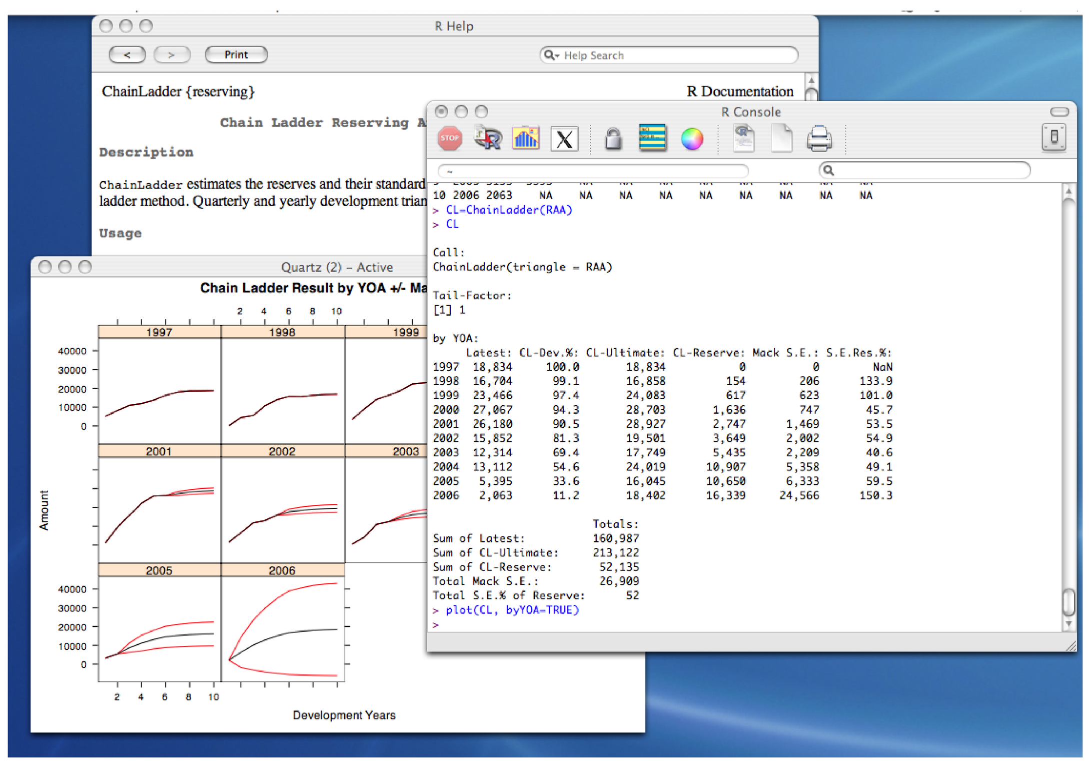
Introduction

Thirty years ago at Bell Labs statisticians developed a computer language for statisticians. They called the language S.

Today there exist two implementations of this language; the open source variant R and the commercially sold product S-Plus by Insightful, see [1, 2, 4, 3].

Within Lloyd's, R is used for producing management information tools, benchmarking, pricing, geographical mapping, data analysis and modelling.

Screenshot of R:



Screenshot of R under Mac OS X

Here we can only present an appetizer. For more information, see [5] and visit `www.r-project.org` or `http://toolkit.pbwiki.com/RToolkit`.

Getting started

R is more or less a command line tool. So is Excel, only in cells.

Instead of ranges, you think in vectors, lists and data frames.

```
x <- 1:10              # create a vector x with 1,2,..., 10
sum(x)                 # sum of x, gives 55 back
y <- rep(2, 100)       # repeat 2 a hundred times
M <- matrix(y, nrow=10) # create a square matrix filled with 2s
M %*% x                # calculate the matrix vector product of M·x
```

Vectorized calculations

The Pearson chi-squared statistic for testing independence in a two-way contingency table is defined as

$$X_P^2 = \sum_{i=1}^{r}\sum_{j=1}^{s}\frac{(f_{ij}-e_{ij})^2}{e_{ij}}, \quad e_{ij} = \frac{f_{i.}f_{.j}}{f_{..}}$$

Instead of two nested for loops we can do this in R, see [6], with a bit of linear algebra knowledge. Say the frequencies f_{ij} are stored in a matrix `f`, than:

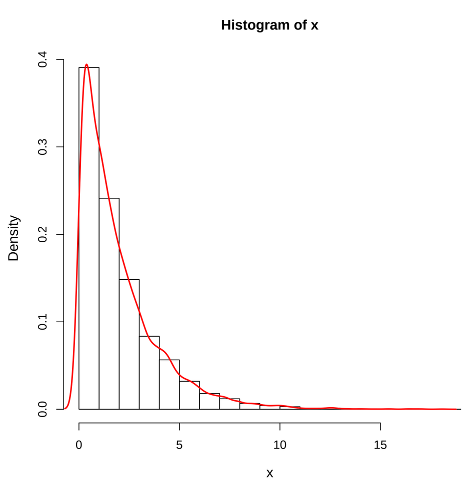
```
fi. <- f %*% rep(1, ncol(f))
f.j <- rep(1, nrow(f)) %*% f
e <- (fi. %*% f.j)/sum(f)
X2p <- sum((f - e)^2/e)
```

will do the trick.

Statistics

Generate exponential distributed random figures, plot their histogram and density curve, and fit a distribution to it:

```
x=rexp(10000, rate=0.5)
hist(x, freq=FALSE); lines(density(x), col="red")
fitdistr(x, "exponential")
     rate
 0.503715630
(0.005037156)
```

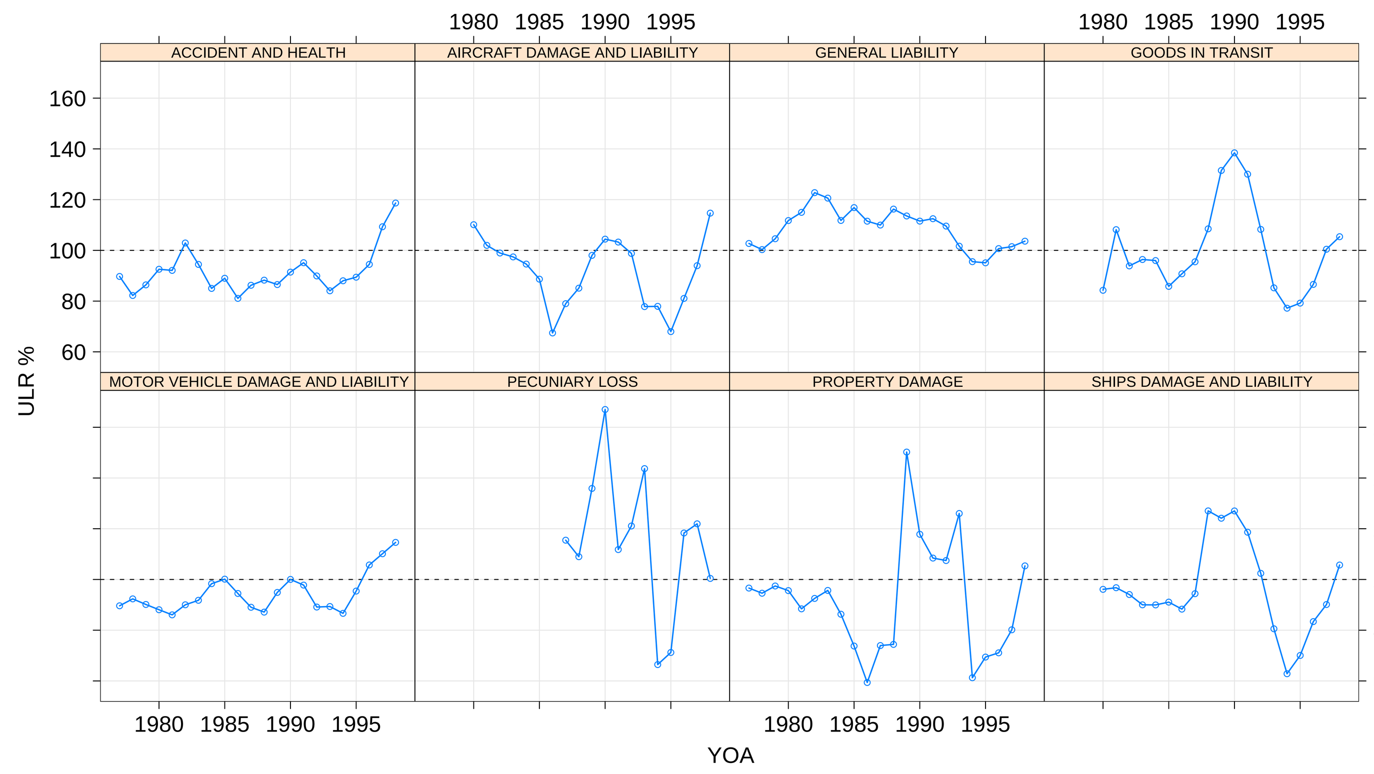


*markus.gesmann@lloyds.com

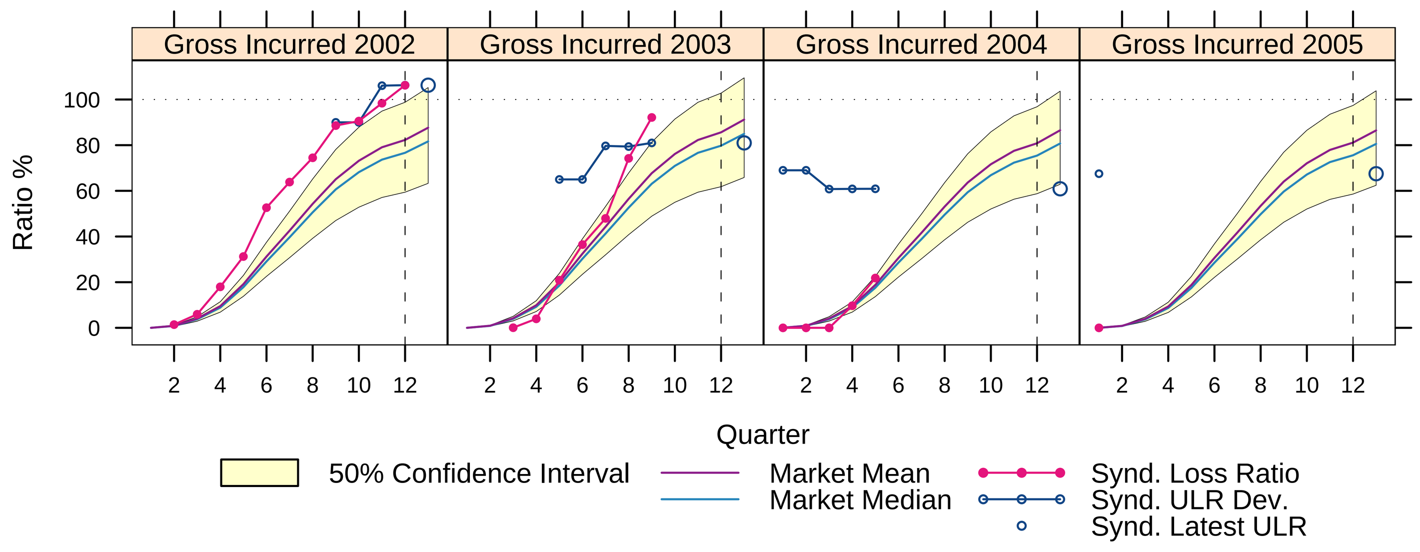
Graphics

Basic example, using the `lattice` package:

```
xyplot(ULR ~ YOA | COB, data=myData)
```



Advanced example:



Data exchange

R offers several options to read and write data, some are

- `read.table`, `write.table` are the workhorses
- the clipboard, e.g. `scan("clipboard")`
- Use `sqlQuery` to send SQL statements to a database via ODBC

Functions for EpiInfo, Minitab, S-PLUS, SAS, SPSS, Stata, Systat exist as well.

R and Excel

The R-Excel Addin offers the ability to access R directly from Excel.

The Pearson chi-squared statistic in R, embedded in Excel VBA:

```
Function rPearson (myData As Range) As Variant
  Call Rput("r.myData", myData)
  Call RInterface.RRun("fi. <- f")
  Call RInterface.RRun(" f.j <- rep(1, nrow(f)) %*% f")
  Call RInterface.RRun("e <- (fi. %*% f.j)/sum(f)")
  Call RInterface.RRun("X2p <- sum((f - e)^2/e)")
  rPearson = reval("X2p")
End Function
```

Functions you'll love

`reshape`	reshapes data between a cross tab (like a triangle) and a long table format.
`merge`	merge two data frames by common columns or rows
`by`	apply a function to each subset of a data set
`optim`	general-purpose optimization, e.g. for curve fitting
`lm, glm`	linear and generalized linear models
`mvnorm`	multivariate normal distribution

References

[1] John M. Chambers and Trevor J. Hastie. *Statistical Models in S*. Chapman & Hall, London, 1992.

[2] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[3] Insightfull, www.insightfull.com. *S-Plus*.

[4] R Development Core Team. *R: A Language and Environment for Statistical Computing*. The R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.

[5] Nigel De Silva. *An Introduction to R: Examples for Actuaries*. Actuarial Toolkit Working Party, version 0.1 edition, 2006. `http://toolkit.pbwiki.com/RToolkit`.

[6] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S. Fourth Edition*. Springer, New York, 2002. ISBN 0-387-95457-0.

Reserving with R

Markus Gesmann*,

Lloyd's, Market Analysis

GIRO Convention, Vienna, 26-29 September 2006

toolkit
RToolkit

Introduction

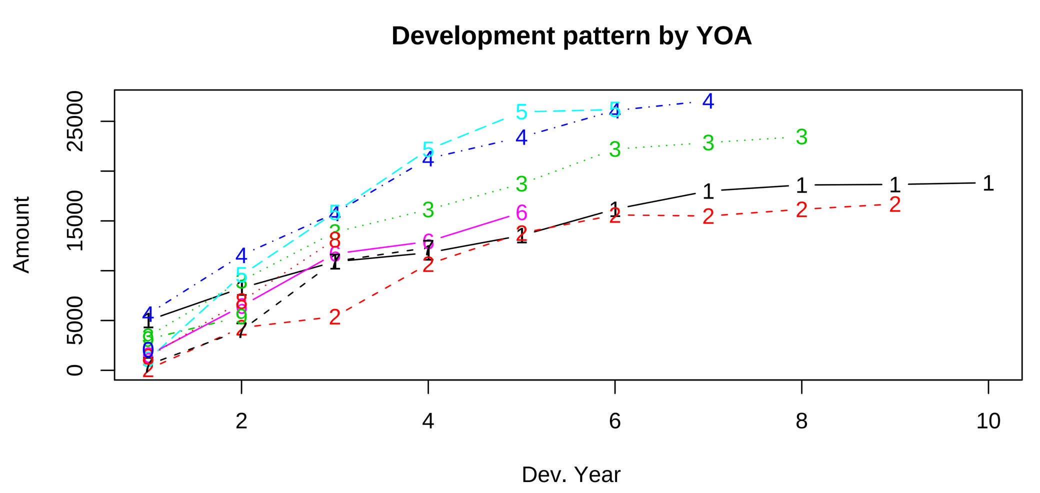
We show some examples on how R can be used in a reserving exercise. The reserving package [1] described here was developed by the author mainly around 2003/2004. At the time it was an exercise which required learning R and understanding actuarial papers [2, 3, 4, 5]. R proved to be a very useful tool to implement these ideas. Here we show some of the results. The reserving package is still in use by the author and also some others who found it useful. The package and further information is available on `http://toolkit.pbwiki.com/RToolkit`.

Triangles

The development of the `reserving` package started from the triangle paradigm. The Mack–, Munich–chain–ladder and bootstrap methods are implemented. The Mack–chain–ladder method shown below:

```
> library(reserving)
> data(RAA) #example data set
> RAA
     V1    V2    V3    V4    V5    V6    V7    V8    V9   V10
1  5012  8269 10907 11805 13539 16181 18009 18608 18662 18834
2   106  4285  5396 10666 13782 15599 15496 16169 16704    NA
3  3410  8992 13873 16141 18735 22214 22863 23466    NA    NA
...
7   557  4020 10946 12314    NA    NA    NA    NA    NA    NA
8  1351  6947 13112    NA    NA    NA    NA    NA    NA    NA
9  3133  5395    NA    NA    NA    NA    NA    NA    NA    NA
10 2063    NA    NA    NA    NA    NA    NA    NA    NA    NA

> matplot(t(RAA), t="b")
```

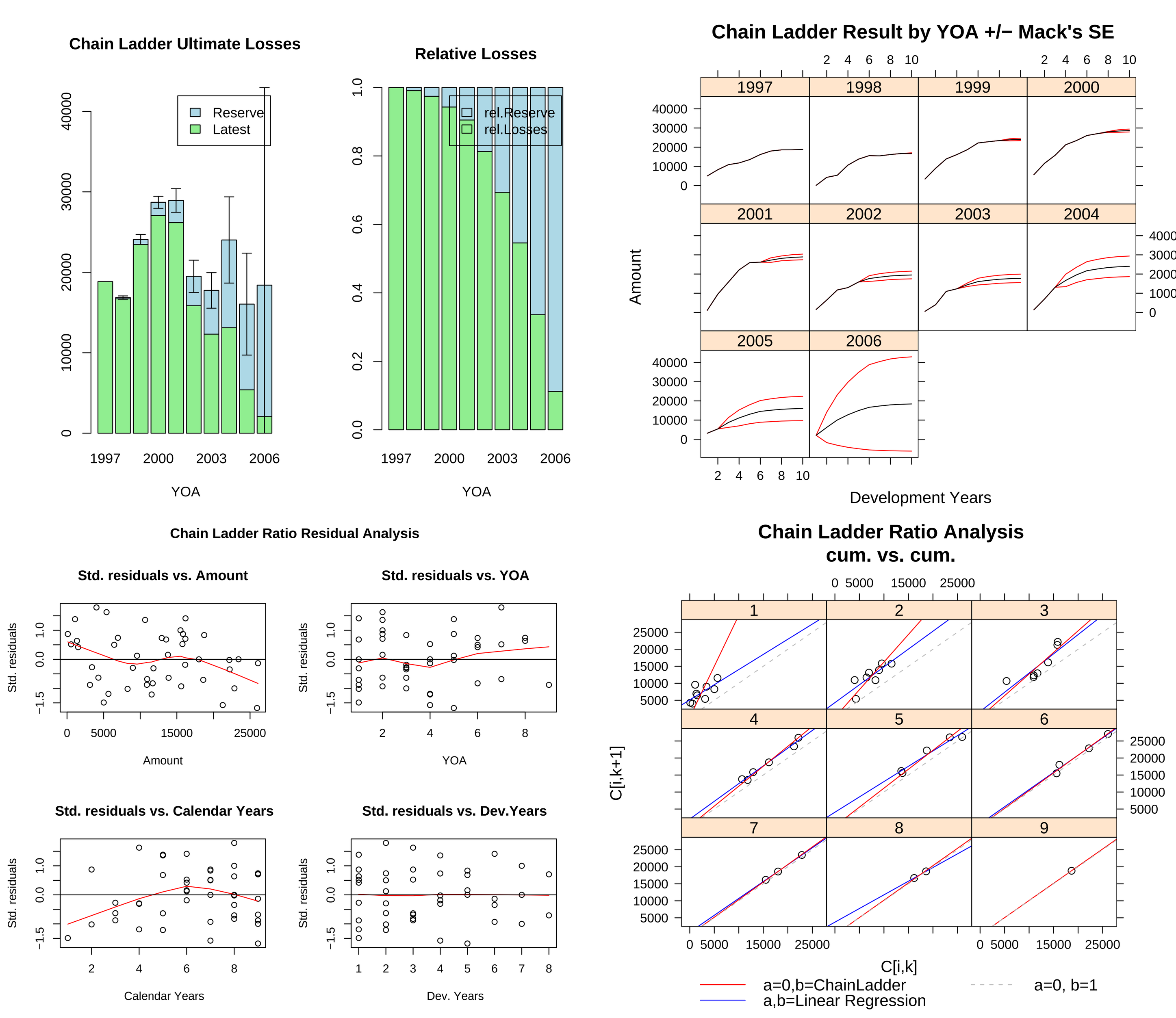



```
> CL=ChainLadder(RAA, YOA=1997:2006)
> CL
 ...
by YOA:
     Latest: CL-Dev.%: CL-Ultimate: CL-Reserve: Mack S.E.: S.E.Res.%:
1997  18,834     100.0       18,834           0          0        NaN
1998  16,704      99.1       16,858         154        206      133.9
1999  23,466      97.4       24,083         617        623      101.0
...
2003  12,314      69.4       17,749       5,435      2,209       40.6
2004  13,112      54.6       24,019      10,907      5,358       49.1
2005   5,395      33.6       16,045      10,650      6,333       59.5
2006   2,063      11.2       18,402      16,339     24,566      150.3

                         Totals:
Sum of CL-Ultimate:      213,122
Total Mack S.E.:          26,909
Total S.E.% of Reserve:       52
```

Diagnostic plots

```
plot(CL); plot(CL, byYOA=TRUE); plotCLResiduals(CL); plotCLRatios(CL)
```



Data frames

Triangles are sometimes a hurdle when working with computers. We transform the triangle into a long format, like

```
> myDATA
   YOA Dev  Cal    x     y
1 1997   1 1998 5012  8269
2 1998   1 1999  106  4285
3 1999   1 2000 3410  8992
4 2000   1 2001 5655 11555
5 2001   1 2002 1092  9565
 ...
```

Fit a model

Now we can directly apply all the statistical modeling functions in R. As pointed out in [5], we can see the chain ladder approach as a weighted linear regression through the origin. Therefore we use the function `lm`, and the chain ladder model becomes a one-liner in R:

```
>  ChainLadder.model=by(myDATA, list(Dev=myDATA$Dev),
                        function(.grp) lm( y ~ x + 0, weights=1/x, data=.grp))
```

We access the chain ladder ratios via:

```
> sapply(ChainLadder.model, coef)
     1.x      2.x      3.x      4.x      5.x      6.x      7.x      8.x      9.x
2.999359 1.623523 1.270888 1.171675 1.113385 1.041935 1.033264 1.016936 1.009217
```

The diagnostic plots suggested that the chain ladder approach is not an appropriate model for the whole data set. Indeed, for the first development period less than 50% of the variation is explained by the model:

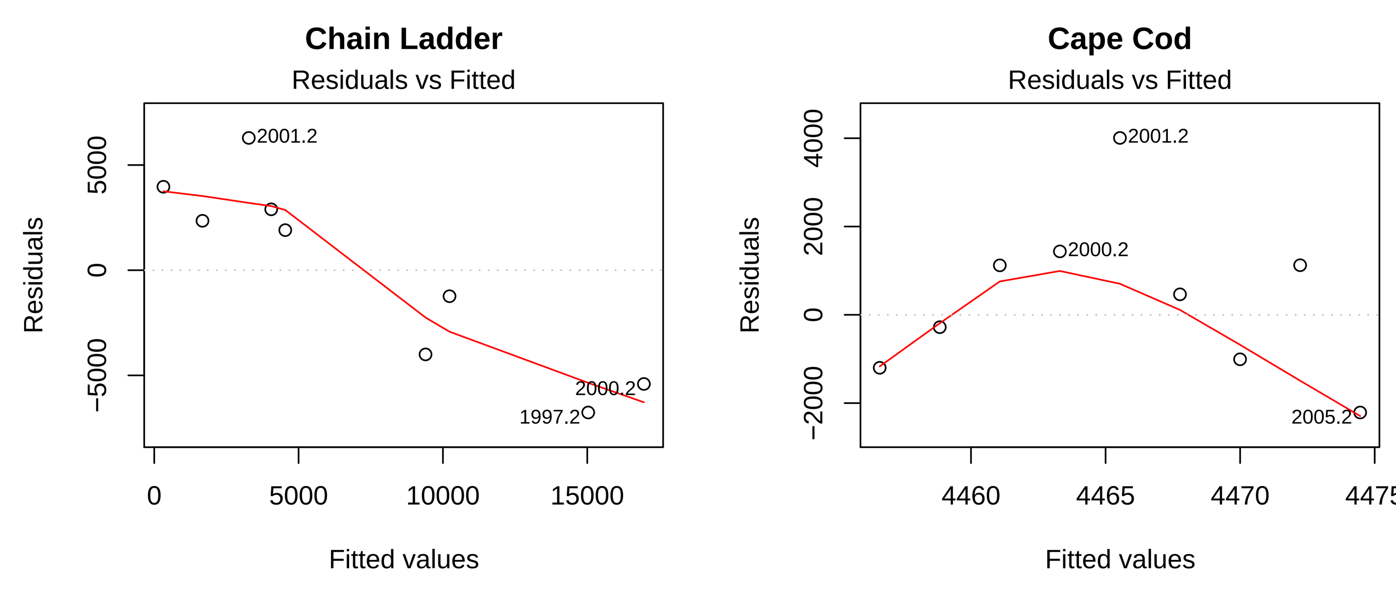
```
>  summary(ChainLadder.model[[1]])
...
Coefficients:
  Estimate Std. Error t value Pr(>|t|)
x    2.999      1.130   2.654   0.0291 *
...
Multiple R-Squared: 0.4682,Adjusted R-squared: 0.4017
```

So let's change the model. All we have to do, is to edit the formula. After some experimenting we will find, that the following model fits well:

```
>  CapeCod.model=by(myDATA, list(Dev=myDATA$Dev),
      function(.grp) lm( I(y-x) ~ YOA + 0, weights=1/x, data=.grp))

> summary(CapeCod.model[[1]])
...
Coefficients:
    Estimate Std. Error t value Pr(>|t|)
YOA   2.2317     0.2122   10.52 5.81e-06 ***
...
Multiple R-Squared: 0.9326,Adjusted R-squared: 0.9241
```

As you can see, we lost the development period as a variable, while the model now only takes the trend along the year of account into account, see as well the residual plots below. Good enough?



Conclusion

R offers a statistical modeling framework which is ideal for reserving. It is quick and easy to find a model which fits your data and not the other way round. Models can be checked and changed with only a few commands. R is free and open source, available for MS Windows, Mac OS and UNIX, so have a go.

References

[1] Markus Gesmann. *reserving: Actuarial tools for reserving analysis*, version 0.1.3 edition, 2003 – 2006. `http://toolkit.pbwiki.com/ReservingPackage`.

[2] 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.

[3] P.D.England and R.J.Verrall. Stochastic claims reserving in general insurance. *British Actuarial Journal*, 8:443–544, 2002.

[4] Gerhard Quarg and Thomas Mack. Munich chain ladder - a reserving methods that reduces the gap between ibnr projections based on paid losses and ibnr projections based on incurred losses. *Blätter DGVFM Deutsche Gesellschaft fur Versicherungs und Finanzmathematik e.V.*, XXVI(4):597 – 630, 2004.

[5] Zehnwirth and Barnett. Best estimates for reserves. *Proceedings of the CAS*, LXXXVII(167), November 2000.

*markus.gesmann@lloyds.com