CAS Working Paper Disclaimer

Working papers are preliminary works in progress that are posted to stimulate discussion and critical comment. The analysis and conclusions set forth are those of the authors. The CAS does not endorse, approve, certify or warrant this material for any purpose, nor does it exercise editorial control over materials posted in this section of the Web Site. Evaluation of the material is the sole responsibility of the user. The CAS, its employees, and agents assume no responsibility for, and expressly disclaim all liability for, any consequences resulting from the use of the information herein.

Third party motor liability ratemaking with R

Spedicato Giorgio Alfredo, Ph.D

December 16, 2011

1 Abstract

GLMs are widely used by the P&C actuarial community. Very well designed softwares exist that help actuaries to implement GLMs within all relevant steps needed to build a commercial TPML tariff from raw data. However few if any freely available documents exist that show how to perform equivalent passages using R. The R software is the most widely known open source statistical package. It is able to perform a wide sprectrum of data analysis techniques due to its diffusion among researchers and practictioners within many scientific fields. This paper wishes to fill this gap, showing the basic of frequency, severity, pure premium modelling, handling a-priori constraints on relativities and how to perform basic no claim discount analysis. A freely available dataset is used allowing readers to repeat the described analyses on their own computers.

2 Introduction

GLMs and their extensions have been used for two decades by actuaries to perform predictive modelling tasks as frequency, severity and lapse probabilities modelling. Increasing computing power and availability of statistical software have helped GLMs to affirm. Software packages specifically tailored to actuarial application of GLMS have been produced like Emblem and Pretium[1]. A comprehensive overview of GLMs actuarial applications in non - life insurance is provided in [Piet de Jong and Gillian Heller, 2008], [Denuit et al., 2007], [Ohlsson and Johansson, 2010].
R [R Development Core Team, 2010] software has been knowing increasing popularity actuaries due to its flexibility and open source nature. The R potential in personal lines pricing has already been discovered by actuaries. On line resources and books exist regarding the use of GLM to price non - life coverages. Moreover packages specifically dedicated to actuarial applications have been published on CRAN. Among these, the actuar package [Dutang et al., 2008] contains routines to fit loss distributions, the ChainLadder package [Gesmann and Zhang, 2011] has been tailored to perform loss reserving analysis in the P&C business, while the lifecontingencies package [Spedicato, 2011] can be used to calculated actuarial present values for life insurance. Standard GLMs can be fit using statistical function bundled in the base R release, since many research fields use GLMs

[1] As Autumn 2011 both these software are sold by Tower Watson consulting firm

modelling. More advanced models, like Tweedie and GAMLSS, can requires dedicated packages (see [Rigby and Stasinopoulos, 2005] and [Zhang, 2011] respectively).

The complete definition of a personal lines tariff, like e.g. a TPML tariff requires however several steps. As deeply described by [Geoff Werner and Claudine Modlin, 2009], an overall rate level shall be determined. Then premium, exposures and losses shall be properly adjusted before beginning the predictive modelling that would lead to obtain relativities. Finally, the relativities arising from predictive modelling shall take into account operational and commercial constraints.
A TPML insurance usually contains relevant examples of such constraints, that shall be considered when performing ratemaking analysis:

1. experience rating features, like a bonus - malus structure or a no - claim discount system (NCD) are usually present. The renewal premium is set forth according to the policyholder claim history. A claim experience variable is present whose relativities are set in advance. The numerical coefficients assigned to this variable are set according to commercial and / or legal constraints. They may significantly differ from the coefficients that would result if they were estimated from raw data. It is therefore important to simulate the evolution of the portfolio within the claim experience rating levels in order to assess whether the overall portfolio premium volume upon renewal will make the charged rate adequate at least considering the aggregated portfolio.

2. commercial restrictions on the relativities of some ratemaking variables. Such restrictions are usually handled using offsets withing the GLMs structure.

A very well prepared tutorial that shows all the passage needed to implement a commercial tariff is the Pretium software manual [Watson, 2010], based on the CAS publication [Anderson et al., 2007]. With respect to the R side, a brief snapshot of R capabilities in non - life pricing has been shown by Chibisi [Chima-Okereke, 2011], while Arthur Charpentier [Charpentier, 2010] prepared a very good document (in French). Charpentier document shows many interesting specific topic regarding the application of R software within R for modelling the frequency, severity and unpaid claim estimate analysis. Nevertheless I have find no paper that fully shows all the passages that bring from raw insurance transaction data toward a commercial tariff using R as December 2011 as [Watson, 2010] does. This paper aims to show how R can be used by pricing actuaries to perform all the fundamental steps needed to build a commercial tariff. The paper will be organized as follows: section 3 will discuss assumptions used in the analysis, section 4 will show how data used in the analysis will be set up, section 5 will show the descriptive analysis, section 6 will show the GLMs fitting process in order to obtain a multiplicative tariff while section 7 will deal with the set up of commercial constraints on coefficients and about how to perform renewal portfolio analysis using a simple NCD system.

All the calculations will be exemplified on a non - proprietary dataset easily available to everyone wants to reply these calculations on his own PC. Therefore the range of analysis will be somewhat limited. In fact this paper will not

year since last filled claim	coefficient
6+	0.25
5	0.40
4	0.50
3	0.60
2	0.70
1	0.80
0	1.00

Table 1: Bonus variable coefficients

deal issues as price optimization (lapse and conversion modelling) or the application of spatial statistics techniques in territorial ratemaking (e.g. spatial smoothing), due to data availability limitation. Nevertheless the R power can be applied with success in these fields also.
Readers will be assumed to be familiar with underlying statistical and actuarial theory regarding GLM and ratemaking. Knowledge of R programming language will also be assumed.

This document has been produced thanks to Sweave [Leisch, 2002] document production facility in order to show how R can be used to produce self-updating and reproducible working documents.

3 Data and assumptions used in the analysis

The data set used in the paper is the ”motorins” data set, bundled within R package Faraway [Faraway, 2011]. The motorins data set represents the Swedish national TPML portfolio experience in 1977.
At that time in Sweden all motor insurance companies applied identical categorization variables to classify customers, and thus their portfolios and their claims statistics could be combined.
A fully description of the dataset can be found in [Hallin and Ingenbleek, 1983]. This dataset have been previously analysed in [Gordon, 2002] with the aim to present Tweedie regression.

Records in the data set represent all possible combinations of four risk classification variables: Kilometres (5 levels), Make (9 levels), Zone (7 levels), and Bonus (7 levels). Each row contains aggregated earned exposures (Insured), number of claims (Claims) and total losses (Payments) of policyholders with the corresponding levels of Kilometres, Make, Bonus and Zone.
The variable Bonus represents the number of years since last filled claim. It represents a No Claim Discount system (NCD). The most relevant issue lies in the fact that its coefficients are not freely estimated from data, but they are fixed in advance according to table 3 values (found in [Cerchiara, 2005]):

As we wish to show how to build a commercial tariff starting from raw data, some assumptions will be made. We will assume that figures in column Payments represent trended, developed to ultimate and adjusted losses (payments plus case reserve and IBNR amounts) for the purposes of this paper. Moreover we will assume formula 1 to calculate the Pr_i commercial premium for risk i-th:

$$Pr_i = \frac{L_i + F}{1 - V - Q} \tag{1}$$

where the L_i term represents a multiplicative loss cost model, the F term represents an allowance for fixed costs, the V term represents the variable expense charge and Q represents an allowance for profit and contingencies. L_i value varies according to risk characteristics. It represents the output of the GLM modelling process, while assigned values for F, V and Q will be 50, 12.5% and 5.0% respectively.
When developing the renewal portfolio analysis, no policyholder is assumed to drop out and the NCD relativities are assumed to be applied upon renewal according to policyholder's claim experience as reported in table 3.

4 Data loading

The code below set up the operating environment and load the dataset

```
> #environment configuration
> #clear the environment
> rm(list=ls())
> #set contrast
> options(contrasts = rep ("contr.treatment", 2))
> #working directory
> setwd('C:\\giorgio lavoro\\universita\\articolo pricing R')
> #See the appendix for a brief description
> #load libraries
> library(faraway)
> library(multcomp)
> #import the data
> data(motorins)
> #show first rows of data
> head(motorins)
```

```
  Kilometres Zone Bonus Make Insured Claims Payment     perd
1          1    1     1    1  455.13    108  392491 3634.176
2          1    1     1    2   69.17     19   46221 2432.684
3          1    1     1    3   72.88     13   15694 1207.231
4          1    1     1    4 1292.39    124  422201 3404.847
5          1    1     1    5  191.01     40  119373 2984.325
6          1    1     1    6  477.66     57  170913 2998.474
```

then we load an utility function file that contains some functions used later in the paper.

```
> #load the utility functions file.
> source("utility.R",echo=TRUE,max.deparse.length=3000)
```

```
> #create a function that return the NCD coefficient given a bonus level
> bonusCoefficient<-function(varBonus)
```

```
+  {
+          out=NULL
+          if(varBonus==1) out=1 else
+          if(varBonus==2) out=0.8 else
+          if(varBonus==3) out=0.7 else
+          if(varBonus==4) out=0.6 else
+          if(varBonus==5) out=0.5 else
+          if(varBonus==6) out=0.4 else
+          if(varBonus==7) out=0.25 else
+         return(out)
+  }

> #create a function that returns the proposed no claim discount class
> #as a function of contract year's number of claims
>
>  proposedNcdClass=function(actual_class, n)
+  {
+         out=NULL
+         if(n==0) out=min(7,actual_class+1) else out=1
+         return(out)
+  }

>  proposedNcdClass2=function(actual_class, n)
+  {
+         out=NULL
+         if(n==0) out=min(7,actual_class+1) else out=max(1, actual_class-2*n)
+         return(out)
+  }

>  oneway<-function(ratemakingFactor)
+  {
+  numClaimsExpr=paste("Claims",ratemakingFactor,sep="~")
+  amountsExpr=paste("Payment",ratemakingFactor,sep="~")
+  exposuresExpr=paste("Insured",ratemakingFactor,sep="~")
+  claims=aggregate(as.formula(numClaimsExpr),data=dataset, FUN="sum")
+  amounts=aggregate(as.formula(amountsExpr),data=dataset, FUN="sum")
+  exposures=aggregate(as.formula(exposuresExpr),data=dataset, FUN="sum")
+  temp<-merge(exposures, claims)
+  temp2<-merge(temp, amounts)
+  temp2$frequency=with(temp2,round(Claims/Insured,2) )
+  temp2$severity=with(temp2,round(Payment/Claims,2) )
+  temp2$burning_cost=with(temp2,round(Payment/Insured,2) )
+  out<-temp2[,c(ratemakingFactor, "Insured","frequency",
+                                  "severity","burning_cost")]
+  return(out)
+  }
```

The following lines recode and label variables in the dataset

```
> #data preparation using descriptions in motorins
> dataset<-transform(motorins,
+ Kilometres=factor(Kilometres, levels=c(1:5),
+ labels=c("1: less than 1000","2: from 1000 to 15 000","3: 15 000 to 20 000",
+ "4: 20 000 to 25 000","5: more than 25 000"),
+ ordered=TRUE),BonusNum=Bonus, Bonus=factor(Bonus, levels=c(1:7),
+ labels=c("1: 0 years no claims", "2: 1 years no claims", "3: 2 years no claims",
+ "4: 3 years no claims","5: 4 years no claims","6: 5 years no claims",
+ "7: 6+ years no claims"), ordered=TRUE),
+ Zone=factor(Zone, levels=c(1:7),
+ labels=c("1: Stockholm, Göteborg, Malmö with surroundings",
+ "2: Other large cities with surroundings",
+ "3: Smaller cities with surroundings in southern Sweden",
+ "4: Rural areas in southern Sweden",
+ "5: Smaller cities with surroundings in northern Sweden",
+ "6: Rural areas in northern Sweden","7: Gotland")),
+ Make=factor(Make, levels=c(1:9)))
```

5 Descriptive statistics

Univariate and bivariate analyses consist in the evaluation of overall frequency, severity and pure premium. Its main purpose lies in verifying the experience data reasonableness using previous experience comparison and professional judgement.

We begin to estimate underlying risk indicators (frequency, severity and burning cost) in the experience data using the following code:

```
> #overall frequency: 5%
> with(dataset, sum(Claims)/sum(Insured))

[1] 0.04756659

> #overall severity: 4955
> with(dataset, sum(Payment)/sum(Claims))

[1] 4955.251

> #pure premium / burning cost: 235.7
> with(dataset, sum(Payment)/sum(Insured))

[1] 235.7044
```

while the "oneway" function (whose code is reported in the appendix) shows the key risk indicators (frequency, severity and burning cost) split by the levels of each classification variables.

```
>        oneway("Bonus")

                  Bonus   Insured frequency severity burning_cost
1  1: 0 years no claims  161026.0      0.12  4526.40       539.40
2  2: 1 years no claims  140308.8      0.08  4770.60       363.16
3  3: 2 years no claims  122555.3      0.06  4911.32       310.26
```

```
4  4: 3 years no claims  110847.9      0.06  4839.82       275.46
5  5: 4 years no claims  136087.1      0.05  4767.10       250.22
6  6: 5 years no claims  253349.5      0.05  4950.17       245.84
7 7: 6+ years no claims 1455037.5      0.03  5211.24       177.37
```

```
>         oneway("Kilometres")
```

```
              Kilometres  Insured frequency severity burning_cost
1      1: less than 1000 805985.6      0.04  4787.37       197.12
2 2: from 1000 to 15 000 803838.7      0.05  4956.77       242.78
3    3: 15 000 to 20 000 476439.1      0.05  5022.30       251.78
4    4: 20 000 to 25 000 172166.2      0.05  5203.84       272.79
5    5: more than 25 000 120782.6      0.06  5171.56       329.86
```

```
>         oneway("Zone")
```

```
                                                    Zone   Insured frequency
1        1: Stockholm, Göteborg, Malmö with surroundings 326149.26      0.07
2                2: Other large cities with surroundings 387642.89      0.05
3 3: Smaller cities with surroundings in southern Sweden 428844.79      0.05
4                      4: Rural areas in southern Sweden 846956.97      0.04
5 5: Smaller cities with surroundings in northern Sweden 119748.86      0.05
6                      6: Rural areas in northern Sweden 251898.10      0.04
7                                             7: Gotland  17971.21      0.03
  severity burning_cost
1  4601.43       326.95
2  4730.79       259.97
3  4858.99       225.91
4  5301.21       199.75
5  4882.52       243.09
6  5387.98       219.50
7  4717.37       162.75
```

```
>         oneway("Make")
```

```
  Make    Insured frequency severity burning_cost
1    1  239379.85      0.05  5248.58       254.82
2    2   50794.11      0.05  5062.35       273.78
3    3   48011.84      0.04  5736.98       220.70
4    4   65693.65      0.03  4235.19       133.13
5    5   53399.85      0.06  4772.72       276.53
6    6  127912.11      0.04  5033.64       183.54
7    7   48634.52      0.04  4693.47       210.38
8    8   23801.05      0.05  6672.47       309.22
9    9 1721585.10      0.05  4898.19       238.56
```

The Zone factor level with the highest exposure is level 4, while the Make factor level with the highest exposure is level 9. R ”treatment” matrix contrast considers the first alphabetical level as the reference level. Multiplicative models coefficients represent therefore implicit relativities with respect to those base level. It is useful to set the base level to the group with most exposure within the same variable, unless custom or other reasons lead the choice of the reference level toward other alternative solutions. In this case we will not alter the reference level for Bonus variable, since custom set the reference level to 1. The reference level is not changed for factor Kilometres since the level with most exposure is already level 1.

```
> dataset$Zone=relevel(dataset$Zone, ref="4: Rural areas in southern Sweden")
> dataset$Make=relevel(dataset$Make, ref="9")
```

6 Predictive modelling: frequency, severity and burning cost modelling

This section will show how models to assess frequency, severity and risk premium can be fit with R. The classical approach used to build classification plans using GLMs is to approximate the burning cost[2] by a multiplicative structure. The loss component of the final rate is obtained by multiplying a base premium for specific coefficients, one for each variable within the classification plan[3]. An estimate of the frequency and the severity of claims is therefore assigned to each row. Then a burning premium is obtained by multiplying these estimates. A multiplicative rating model is therefore estimated on the burning cost in order to obtain the expected loss component according to the policyholder's characteristics.

The frequency of claims is usually fit by an overdispersed Poisson GLM (ODP). A log-linear Gamma GLM will be fit to asses the severity of claims, using the expected cost (incurred amount divided by number of claims) as dependend variable and the number of claims as weights. The previous models will be combined in order to estimate a fitted risk premium model using two approaches:

1. A first model will be estimated to create a multiplicative ratemaking structure. This model (the free burning cost model) will return the pure premium according to the policyholder's characteristics.

2. Another model (the constrained burning cost model) will be estimated following the same base criterion. However some variables will have their relativities fixed in advance.

These two last models will be estimated by a Gamma GLM with a logarithmic link.

6.1 Frequency modelling

The number of claims will be modelled by a log-linear count GLM (Poisson model), using the logarithm of total insured years (Insured) as offset.

[2] We will the word burning cost as equivalent of risk premium and pure premium.

[3] Some adjustments to this algorithm shall be made in case of interactions between the classification variables. However, the tariff structure we are building does not contain interactions at all.

```
> #Poisson GLM
> freqModel<-glm(Claims~Kilometres+Zone+Bonus+
+ Make,offset=log(Insured), data=dataset,family="quasipoisson")
> anova(freqModel, test="Chisq") #test III

Analysis of Deviance Table

Model: quasipoisson, link: log

Response: Claims

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                        1796      33694
Kilometres  4   1509.0      1792      32185 < 2.2e-16 ***
Zone        6   6070.8      1786      26115 < 2.2e-16 ***
Bonus       6  22182.2      1780       3932 < 2.2e-16 ***
Make        8   1442.9      1772       2489 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type III tests invoked with last line of code indicate that all inserted factors are statistically significant.

Figure 1 shows marginal effects plots of the frequency model. It is worth to inspect the monotone relationships between the marginal claim frequency coefficients and the levels of either Bonus or Kilometres factors.

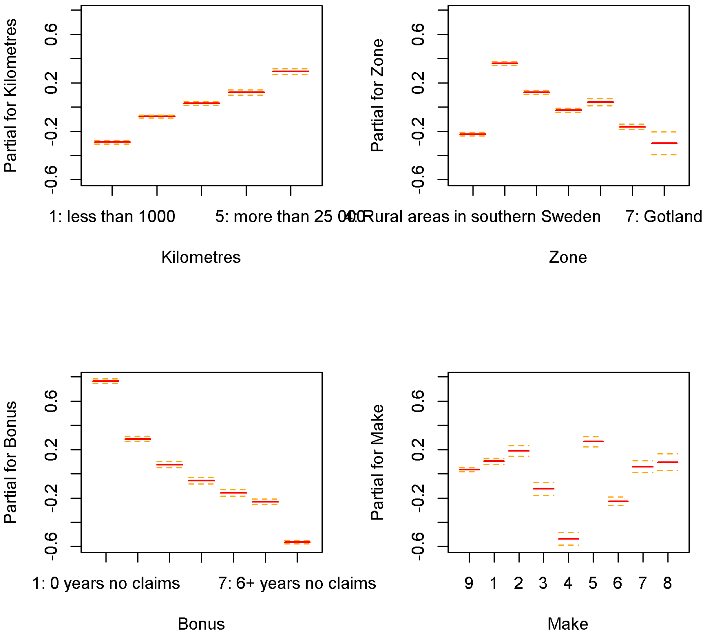
The analysis of figure 1 shows that confidence bands of some levels of Make and Zone factors overlap. The identification of which levels are characterized by statistically equivalent coefficients is a worth step. Aggregating statistically equivalent levels within the same factor is a good practice in GLM modelling. It helps to create models with greater consinstency and robustness. Package multcomp [Hothorn et al., 2008] allows multiple comparisons within R framework. Even if we will not aggregate levels with statistically equivalent coefficients, we show the code that exemplifies how similar levels can be identified. The inspection of reported log shows that levels 7,8 and 1 within the Make factor seem statistically equivalent, for example.

```
> #performs post hoc analysis on the make factor using multcomp package glht
> freqModelMakeMultComp<-glht(freqModel, linfct = mcp(Make = "Tukey"))
> summary(freqModelMakeMultComp)

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: glm(formula = Claims ~ Kilometres + Zone + Bonus + Make, family = "quasipoisson",
```



paper-fig1.png

Figure 1: Frequency marginal effects plot

```
    data = dataset, offset = log(Insured))

Linear Hypotheses:
           Estimate Std. Error z value Pr(>|z|)
1 - 9 == 0  0.06960    0.01229   5.662  < 0.001 ***
2 - 9 == 0  0.15598    0.02404   6.489  < 0.001 ***
3 - 9 == 0 -0.15642    0.02918  -5.361  < 0.001 ***
4 - 9 == 0 -0.57114    0.02773 -20.597  < 0.001 ***
5 - 9 == 0  0.23111    0.02262  10.216  < 0.001 ***
6 - 9 == 0 -0.26164    0.01859 -14.072  < 0.001 ***
7 - 9 == 0  0.02489    0.02682   0.928  0.98868
8 - 9 == 0  0.06130    0.03761   1.630  0.75405
2 - 1 == 0  0.08638    0.02622   3.294  0.02346 *
3 - 1 == 0 -0.22601    0.03099  -7.294  < 0.001 ***
4 - 1 == 0 -0.64074    0.02987 -21.448  < 0.001 ***
5 - 1 == 0  0.16151    0.02498   6.465  < 0.001 ***
6 - 1 == 0 -0.33124    0.02145 -15.440  < 0.001 ***
7 - 1 == 0 -0.04471    0.02882  -1.551  0.80085
8 - 1 == 0 -0.00830    0.03902  -0.213  1.00000
3 - 2 == 0 -0.31240    0.03716  -8.406  < 0.001 ***
4 - 2 == 0 -0.72712    0.03640 -19.978  < 0.001 ***
5 - 2 == 0  0.07513    0.03241   2.318  0.29479
6 - 2 == 0 -0.41762    0.02980 -14.013  < 0.001 ***
7 - 2 == 0 -0.13109    0.03545  -3.698  0.00566 **
8 - 2 == 0 -0.09468    0.04407  -2.148  0.39875
4 - 3 == 0 -0.41472    0.04002 -10.364  < 0.001 ***
5 - 3 == 0  0.38752    0.03638  10.653  < 0.001 ***
6 - 3 == 0 -0.10522    0.03409  -3.086  0.04447 *
7 - 3 == 0  0.18131    0.03911   4.636  < 0.001 ***
8 - 3 == 0  0.21771    0.04703   4.629  < 0.001 ***
5 - 4 == 0  0.80225    0.03535  22.693  < 0.001 ***
6 - 4 == 0  0.30950    0.03283   9.428  < 0.001 ***
7 - 4 == 0  0.59603    0.03821  15.600  < 0.001 ***
8 - 4 == 0  0.63244    0.04655  13.587  < 0.001 ***
6 - 5 == 0 -0.49274    0.02865 -17.200  < 0.001 ***
7 - 5 == 0 -0.20622    0.03453  -5.972  < 0.001 ***
8 - 5 == 0 -0.16981    0.04343  -3.910  0.00260 **
7 - 6 == 0  0.28653    0.03207   8.935  < 0.001 ***
8 - 6 == 0  0.32294    0.04152   7.777  < 0.001 ***
8 - 7 == 0  0.03640    0.04574   0.796  0.99596
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

6.2 Severity modelling

We will perform similar steps in order to model the severity of claims

```
> dataset$averageCost=with(dataset, Payment/Claims)
> #Gamma GLM on severity
> sevModel<-glm(averageCost~Zone+Make+Bonus+Kilometres,
+ weights=Claims, data=dataset, family=Gamma(link="log"))
> anova(sevModel, test="Chisq") #test III

Analysis of Deviance Table

Model: Gamma, link: log

Response: averageCost

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                        1796     5417.7
Zone        6   402.23      1790     5015.5 < 2.2e-16 ***
Make        8   242.34      1782     4773.2 1.809e-14 ***
Bonus       6   225.86      1776     4547.3 1.834e-14 ***
Kilometres  4    20.73      1772     4526.6    0.1345
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #Kilometres factor not significant
> sevModel<-update(sevModel, ~.-Kilometres)
> anova(sevModel, test="Chisq") #test III

Analysis of Deviance Table

Model: Gamma, link: log

Response: averageCost

Terms added sequentially (first to last)


      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                   1796     5417.7
Zone   6   402.23      1790     5015.5 < 2.2e-16 ***
Make   8   242.34      1782     4773.2 2.619e-14 ***
Bonus  6   225.86      1776     4547.3 2.610e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #we decide to remove also Bonus even if statistically significant, as
> #coefficients profile seem judgmentally randomic
> sevModel<-update(sevModel, ~.-Bonus)
```

Figure 2 shows the marginal effect plot of severity modelling. Only Zone and Make are identified as significant.

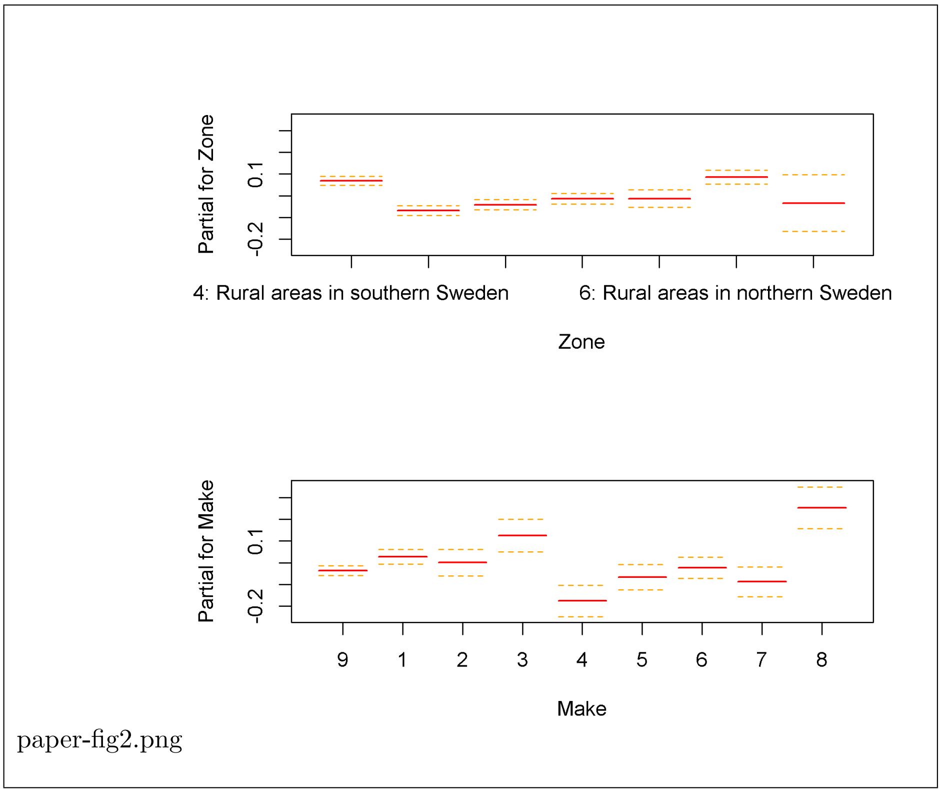


Figure 2: Severity marginal effects plot

6.3 Burning cost modelling

The first step consists in determining the burning cost for each class of insureds.

```
> #calculate fitted frequency and fitted severity
> fittedFrequency=predict(object=freqModel, newdata=dataset,type="response")
> fittedSeverity=predict(object=sevModel, newdata=dataset,type="response")
> #add those columns on main dataset for convenience
> dataset<-transform(dataset,
+                 fittedFrequency=fittedFrequency,
+                 fittedSeverity=fittedSeverity,
+                 purePremium=(fittedFrequency*fittedSeverity)/Insured
+ )
> #check overall balance
> with(dataset, sum(purePremium*Insured))

[1] 560787950

> with(dataset, sum(Payment))

[1] 560790681

> #the difference between actual losses and losse amount is not material
> #add bonus malus coefficient
> dataset<-transform(dataset,
+         ncdCoeff=sapply(as.numeric(Bonus),bonusCoefficient))
```

Then a non - constrained risk premium model is fit by a Gamma GLM, using a multiplicative rating structure.

```
> burnCostFreeModel<-glm(purePremium~Zone+Bonus+Make+Kilometres,
+ weights=Insured, data=dataset, family=Gamma(link="log"))
```

The residual deviance is close to zero in the free burning cost model. In fact the risk premiums values do not represent a truly outcome of a stochastic variable. They are indeed deterministically determined by the product of fitted frequency and fitted severity. Therefore a multiplicative model is able reverse engineer the risk premium model without uncertainty.

```
> #this is the burning cost model with bonus malus coefficients constrained
> burnCostConstrModel<-glm(purePremium~Zone+Make+Kilometres,
+         weights=Insured, data=dataset, family=Gamma(link="log"), offset=log(ncdCoeff))
> #add the risk premium to the dataset
> dataset$risk_premium=predict(burnCostConstrModel,
+         newdata=dataset, type="response")
> #check the balance
> #actual total losses
> with(dataset, sum(Payment))

[1] 560790681

> #charged total losses
> with(dataset, sum(Insured*risk_premium))
```

```
[1] 588050441
```

```
> #5% delta
```

We can therefore compare the free and the constrained burning cost models with respect to estimated coefficients:

```
> #obtain the coefficients for both model
> coefFree<-(coef(burnCostFreeModel))
> coefConstr<-coef(burnCostConstrModel)
> #create a pretty table and exponentiate coefficient
> tableCoefFree<-data.frame(levels=names(coefFree),
+                 coefficientsFree=exp(as.numeric(coefFree)))
> tableCoefConstr<-data.frame(levels=names(coefConstr),
+                 coefficientRestricted=exp(as.numeric(coefConstr)))
> tableCoeff<-merge(x=tableCoefFree,y=tableCoefConstr)
> #output the table
> print(tableCoeff,digits=2)
```

```
                                                       levels coefficientsFree
1                                                 (Intercept)           445.69
2                            Kilometres2: from 1000 to 15 000             1.24
3                               Kilometres3: 15 000 to 20 000             1.38
4                               Kilometres4: 20 000 to 25 000             1.51
5                               Kilometres5: more than 25 000             1.79
6                                                       Make1             1.14
7                                                       Make2             1.21
8                                                       Make3             1.01
9                                                       Make4             0.49
10                                                      Make5             1.22
11                                                      Make6             0.78
12                                                      Make7             0.97
13                                                      Make8             1.42
14        Zone1: Stockholm, Göteborg, Malmö with surroundings             1.56
15                Zone2: Other large cities with surroundings             1.26
16 Zone3: Smaller cities with surroundings in southern Sweden             1.12
17 Zone5: Smaller cities with surroundings in northern Sweden             1.20
18                      Zone6: Rural areas in northern Sweden             1.08
19                                             Zone7: Gotland             0.84
   coefficientRestricted
1                 435.68
2                   1.23
3                   1.38
4                   1.52
5                   1.80
6                   1.16
7                   1.24
8                   1.04
9                   0.47
10                  1.24
11                  0.78
```

```
12                  0.99
13                  1.46
14                  1.54
15                  1.25
16                  1.11
17                  1.19
18                  1.08
19                  0.84
```

The coefficient output shown before makes clear that unconstrained and constrained burning cost coefficients are close in most cases. Moreover the premium for a policyholder making no more than 1000 KM per year, living in Zone 4, driving a Make 9 car, and whose NCD level being 1 is equal to 435.68.[4]

An alternative approach would have involved the Tweedie regression. The classical modelling approach models separately frequency and severity of claims. Their expected value are then combined and a final GLM is fitted on their product to generate a multiplicative tariff (taking into account constraints on coefficients using offset parameters, whether necessary). Tweedie regression converserly models directly the total loss, assuming total loss being the outcome of a compound Poisson process (see [Dunn and Smyth, 2005] for details). The cpml R package [Zhang, 2011] implements Tweedie regression in R. The code reported below shows how a Tweedie regression can be used to fit both an uncontrained and a constrained burning cost model.

```
> #load library
> library(cplm)
> #fit the unconstrained burning cost model
> tweedieUnconstrained<-cpglm(Payment˜Bonus+Zone+Make+Kilometres+
+         offset(log(Insured)), link="log" , data=dataset)
> #fit the constrained burning cost model
> tweedieConstrained<-cpglm(Payment˜offset(log(ncdCoeff))+Zone+Make+
+         Kilometres+offset(log(Insured)), link="log" , data=dataset)
>
```

7 Calculating new business and renewal commercial premiums

7.1 New business premium

A commercial tariff usually takes into accounts loads for fixed and variable expenses and a profit & contingency, in addition to the insured loss cost component. A multiplicative balancing constant will finally grant that the actual experienced losses will be balanced by the amounts obtained applying the rating structure to the portfolio.

```
> #determine the balancing constant
> balancing_constant=with(dataset, sum(Payment)/sum(Insured*risk_premium))
```

[4] Before ad adjustment coefficient determined in the following paragraph.

```
> #pre set constants for fixed, variable expense and profit loads
> F=50
> V=0.125
> Q=0.05
> #calculate the new business commercial tariff
> dataset$new_business_premium=(predict(burnCostConstrModel,
+ newdata=dataset, type="response")*balancing_constant+F)/(1-V-Q)
> #get the loss ratio
> with(dataset, sum(Payment)/sum(new_business_premium*Insured))
```

```
[1] 0.68062
```

7.2 Renewal portfolio evolution analysis

NCD relativities are usually set according to commercial considerations and / or regulatory constraints. The loss cost component of the commercial tariff applied at renewal shall be compared with the projected burning cost of the portfolio during the forthcoming period. In fact the commercial tariff often needs to be rebalanced in order to offset a shortfall with respect to the projected losses during the renewal period.

Then we simulate the NCD evolution using following steps:

1. generating a sample portfolio of policies for which the claim experience before renewal and proposed renewal premium will be simulated.
2. simulating the number of claims per policy prior renewing.
3. assigning the corresponding NCD level according to previous period experience.
4. rating the renewal proposed premium using the constrained burning premium model in order to estimate the premium inflows.
5. re - rating the policy using the unconstrained risk premium model in order to estimate the projected losses.
6. evaluating the shortfall between these evaluations.

Sampling is necessary in order to avoid long computational time when simulating an entire portfolio NCD evolution and loss experience. A sample of five thousands policyholders that reflects the policyholders' distribution of the main dataset is generated by the following code. The rows of motorins dataset do not represent a single policy transaction history, in fact. They represent the aggregated experience of all policyholders defined by the unique ratemaking variables combination defined in the row.

```
> #consider only ratemaking factors
> data2Sample<-dataset[,c("Kilometres","Zone","Bonus","Make","Insured")]
> #simulate the NCD evolution on 5K insureds
> toSample=5000
> #this code will return a dataset sized 5k that reflects the risk profiles
> #proportion found in the original data set
>
> data2Sample$proportion=with(data2Sample, Insured/sum(Insured))
> for(i in 1:dim(data2Sample)[1]){
+         profile_dim=rbinom(1,prob=data2Sample$proportion[i],size=toSample)
+         if(profile_dim>0) {
+                 out=data2Sample[rep(i,profile_dim),c("Kilometres","Zone",
+                                                 "Bonus","Make","Insured")]
+                 row.names(out)<-NULL
+         if(!exists("renewals")) renewals=out else renewals=rbind(renewals, out)
+         }
+ }
> #make every insured having one years of experience
> renewals$Insured=1
> #determine each insured underlying frequency
> renewals$lambda=predict(freqModel, newdata=renewals, type="response")

> #simulate one year claims prior renewal
> renewals$claims=with(renewals, mapply(FUN="rpois", n=1, lambda=lambda))
> #simulate corresponding NCD evolution at renewal
> renewals$BonusNum=as.numeric(renewals$Bonus)
> renewals$NewBonus=with(renewals,
+ mapply(FUN="proposedNcdClass", actual_class=BonusNum, n=claims))
> renewals$Bonus=renewals$NewBonus
> renewals<-renewals[,c("Kilometres","Zone","Bonus","Make", "Insured")]
> renewals<-transform(renewals, Bonus=factor(Bonus, levels=c(1:7),
+ labels=c("1: 0 years no claims", "2: 1 years no claims",
+                 "3: 2 years no claims","4: 3 years no claims",
+                 "5: 4 years no claims","6: 5 years no claims",
+                 "7: 6+ years no claims"), ordered=TRUE))
> #assign coefficients of the new NCD scale at renewal
> renewals$ncdCoeff=sapply(as.numeric(renewals$Bonus), "bonusCoefficient")
```

The code below allows a comparison of the expected total loss amount to the multiplicative burning cost component charged when the commercial tariff is applied (using the constrained burning cost model).

```
> #evaluate charged cost
> renewals$burncost_free=predict(burnCostFreeModel, renewals, "response")
> #evaluate prospective cost
> renewals$burncost_constr=balancing_constant*predict(burnCostConstrModel,
+ renewals, "response")
> #determine offset coefficient
> adjustmentCoeff=with(renewals, sum(burncost_free)/sum(burncost_constr))
```

Therefore the multiplicative component of the commercial tariff formula 1 shall be multiplied by a factor 1.01 to grant the premium adequacy of the insured portfolio upon renewal[5]. The adjustment seems low, but actual NCD relativities applied have been found very close to the unconstrained burning cot model

[5] inflation rate has been assumed equal to 0%

indicated relativities. Often the distance between indicated and commercial coefficients for a NCD systems is far greater.

8 Conclusions

This paper has shown how all most relevant issues needed to build a TMPL tariff can be handled by the open source R software. R capabilities could be applied also to more sofisticated issues like price optimization (lapse and conversion modelling) or zoning classification using spatial smoothing. However no free data has been found to show how to perform these tasks by the R software to a non restricted audience.

The R software advantages lies in the wide spectrum of statistical analysis that can be perfomed and, of course, in its price since it is available at no charge. At the same time the learning curve for R software is steeper than the learning curve of most commercial packages. Moreover R software is still weaker than Emblem when compared with computational speed and SAS is deemed more fast and reliable when huge amount of data needs to be managed. The situation is expected to change toward significant R improvements since a great community of data analyst and computational statistician works to improve R.

References

[Anderson et al., 2007] Anderson, D., Feldblum, S., Modlin, C., Schirmacher, D., Schirmacher, E., and Thandi, N. (2007). A practitioner's guide to generalized linear models. Technical report, Casualty Actuarial Society.

[Cerchiara, 2005] Cerchiara, R. R. (2005). Bonus malus systems and no claim discount: effects on the solvency of a non - life insurance company. In *Atti del XII convegno di teoria del rischio.*

[Charpentier, 2010] Charpentier, A. (2010). Statistique de l'assurance, stt 6705v.

[Chima-Okereke, 2011] Chima-Okereke, C. (2011). R in actuarial pricing teams. ppt.

[Denuit et al., 2007] Denuit, Marechal, Pitrebois, and Wahiln (2007). *Actuarial Modelling of Claim Counts: Risk Classification, Credibility and Bonus-Malus Systems.*

[Dunn and Smyth, 2005] Dunn, P. and Smyth, G. (2005). Series evaluation of tweedie exponential dispersion model densities. *Statistics and Computing*, 15:267–280. 10.1007/s11222-005-4070-y.

[Dutang et al., 2008] Dutang, C., Goulet, V., and Pigeon, M. (2008). actuar: An r package for actuarial science. *Journal of Statistical Software*, 25(7):38.

[Faraway, 2011] Faraway, J. (2011). *faraway: Functions and datasets for books by Julian Faraway.* R package version 1.0.5.

[Geoff Werner and Claudine Modlin, 2009] Geoff Werner and Claudine Modlin (2009). *Basic Ratemaking.*

[Gesmann and Zhang, 2011] Gesmann, M. and Zhang, Y. (2011). *ChainLadder: Mack, Bootstrap, Munich and Multivariate-chain-ladder Methods.* R package version 0.1.5-0.

[Gordon, 2002] Gordon, K. (2002). Fitting tweedie's compound poisson model to insurance claims data: dispersion modelling. *Astin Bulletin*, 32(1):143–157.

[Hallin and Ingenbleek, 1983] Hallin, M. and Ingenbleek, J.-F. (1983). The swedish automobile portfolio in 1977: a statistical study. ULB Institutional Repository 2013/1997, ULB – Universite Libre de Bruxelles.

[Hothorn et al., 2008] Hothorn, T., Bretz, F., and Westfall, P. (2008). Simultaneous inference in general parametric models. *Biometrical Journal*, 50(3):346–363.

[Leisch, 2002] Leisch, F. (2002). Sweave, part I: Mixing R and LaTeX. *R News*, 2(3):28–31.

[Ohlsson and Johansson, 2010] Ohlsson, E. and Johansson, B. (2010). *Non-Life Insurance Pricing with Generalized Linear Models.* Eaa Series: Textbook. Springer Verlag.

[Piet de Jong and Gillian Heller, 2008] Piet de Jong and Gillian Heller (2008). *Generalized linear models for insurance data*. Cambridge University Press, New York, first edition edition.

[R Development Core Team, 2010] R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[Rigby and Stasinopoulos, 2005] Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape,(with discussion). *Applied Statistics*, 54:507–554.

[Spedicato, 2011] Spedicato, G. A. (2011). *Lifecontingencies: an R package to perform life contingencies calculations*. R package version 0.0.4.

[Watson, 2010] Watson, T. (2010). *Pretium manual*. Tower Watson, 3.1 edition.

[Zhang, 2011] Zhang, W. (2011). *cplm: Monte Carlo EM algorithms and Bayesian methods for fitting Tweedie compound Poisson linear models*. R package version 0.2-1.