Pricing Insurance Contracts with R

R in Insurance conference
CASS Business School
London 15th 2013

Giorgio Alfredo Spedicato
Ph.D C.Stat ACAS

15th July 2013

Cass Business School
CITY UNIVERSITY LONDON

Table of contents

Outline

R for insurance applications

- R statistical system, [R Development Core Team, 2012], is becoming the leading software for performing statistical analysis in Academics. Also, in business application it is now a reliable competitor to affirmed data analysis software like SAS [SAS Institute Inc., 2011], SPSS [IBM Corp, 2012] and Matlab [The MathWorks Inc., 2010].
- Packages specifically devoted to actuarial applications have been available for some years especially for non - life business applications.
- This presentation will show how R can be successfully applied to price insurance contracts. Life insurances, general insurances personal lines and XL reinsurance treaty rate making will be shown.

What does pricing mean?

- Insurance pricing means setting the cost of the future uncertain benefits the insurer would provide to its policyholders.
- An insurance rate, P, is an estimate of the expected value of future costs, that provides for all costs associated with the transfer of risk, according to CAS ratemaking principles [Casualty Actuarial Society, 1988].

$$PV\left(P\right) = PV\left(\tilde{L} + \tilde{E} + Q\right) \tag{1}$$

What this presentation shows

- Costs associated with the transfer of risk that premium $\tilde{P}$ allows for are: the future claims cost $\tilde{L}$, allocated expenses E as well as profit and contingency loads Q as Equation 1 expresses.
- In the examples that follow, in addition to future losses / insurance benefits $\tilde{L}$, the premium will have a provision for 5% of variable expenses. Only for life insurances future insurance benefit will be discounted by 2.5% (PV).
- The profit load percentage will be calculated in order the insurer to earn 15% on allocated capital. Allocated capital will be calculated using a VaR approach: $C_t = \tilde{L}_{.99} - E\left[\tilde{L}\right]$, being $\tilde{L}$ the portfolio benefit distribution and allowed within the profit and contingency load $\tilde{L}$ as Equation 2 displays.

$$CoC_{\text{per contract}} = \frac{0.15 * \left(\tilde{L}_{.99} - E\left[\tilde{L}\right]\right)}{Exposures} \tag{2}$$

What this presentation shows

- All examples will show that R tools can be used to model not only the cost component, $\tilde{L}$, but also to appreciate the underlying cost of capital above defined on which the profit and contingency load will be calculated.
- Full code for the three example will be shown in the Appendix.

Outline

Life insurance pricing

- Pricing life insurance contracts requires demographical assumptions to be set out since future cash flows are contingent on policyholders life status (life contingencies).
- Similarly, financial assumptions are required as well since the long duration of life contracts allows the insurer to invest premiums and reserves proceeds.
- The example that follows is worked out with R package lifecontingencies, [Spedicato, 2013]. It provides a full set of functions that life actuaries could use in their professional activity.

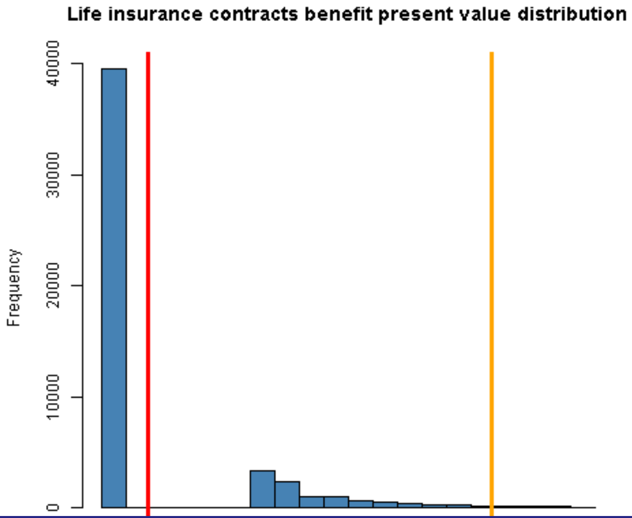
Life insurance example

- A term life insurance is to be sold to 1,000 individuals each aged 25, term is 40 years. Face value is 100,000, single premium.
- Assume SoA illustrative life table to hold, interest rate 2.5%
- Acquisition and handling expenses are: 3% of premium. Profit load is 15% of the portfolio benefit distribution.

Life insurance example

- lifecontingencies package [Spedicato, 2013] allows the actuary to model the full distribution of underlying life contingency random variable, $\tilde{L}$, not only its classical actuarial present value $100000 * A^{1}_{25:\overline{40|}}$. This allows to straightly incorporate capital consideration in life insurance pricing.
- Figure 1 shows the distribution of the benefits associated to a single policy.
- Table 1 displays key figures for the life insurance pricing exercise exemplified.

Life insurance example



Life insurance example

size	1000.00
capital	100000.00
duration	40.00
age	25.00
expense ratio	0.03
final premium	9994.42

Table : Life insurance example key figures

Outline

Introduction

- Personal lines data bases contain millions of record regarding policyholders' contracts and associated claims.
- It is customary to buld predictive models on the frequency, severity and (or directly) the risk premium of the coverages. Currently, log-linear GLMs are the most used models.
- Often final relativities diverge from modelled ones allowing for regulatory and / or marketing constraints.

Introduction

- Standard GLM can be fit using *glm* function within base R.
- Tweedie GLM used to fit directly risk premium is implemented in *cplm* package, [Zhang, 2012].
- In case needed, R *parallel* package can reduce the time to simulate the portfolio aggregated total loss

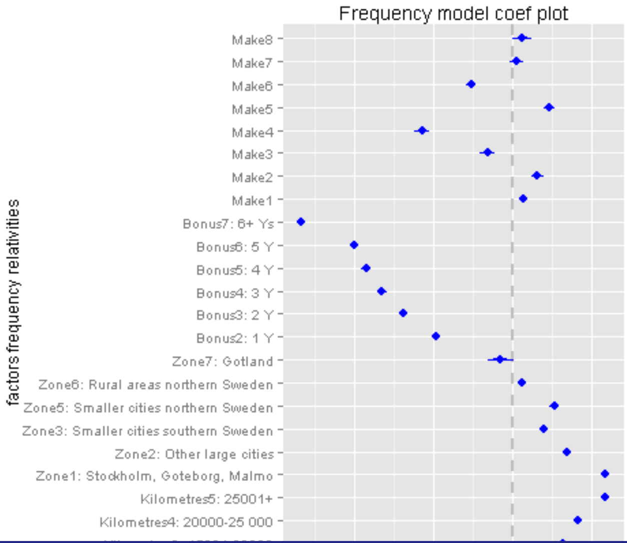
The MTPL example

- The data comes from *faraway* package [Faraway, 2011]. A worked actuarial example on this dataset can be found in [Spedicato, 2012].
- It contains aggregate exposures, number and total MTPL losses by level of classification variable of the Swedish MTPL insurance bureau (70s data). While very old, it is however very didactical.
- Models on the frequency and the severity of losses will be fit. Similarly, the distribution of insured benefits will be modelled in order to incorporate cost of capital consideration in the final rate.

Frequency modelling

- Frequency is modelled by an overdispersed Poisson model using $\ln$ *Insured* as offset term.
- Figure 2 shows the coefficient plot as given by coefplot [Lander, 2013] package.
- The interpretation is quite natural for some coefficients. For example, frequency decreases by Bonus level and increases by Average Mileage. Vehicle make and territory relativities are also displayed.

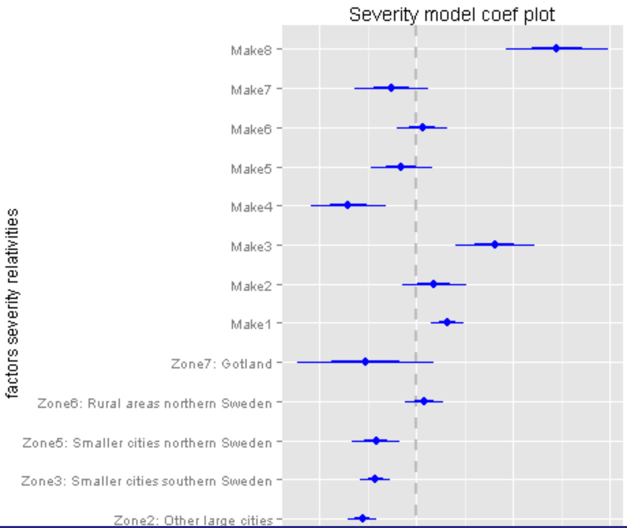
Frequency modelling



Severity model

- Claim cost is modelled by a Gamma log-linar model
- The severity is used as dependent variable and number of claims is used as weight.

Severity modelling



Pure premium modelling

- Fitted frequency and severity are saved for each row in order the pure premium to be calculated. Two Gamma GLM models will be fit on such calculated value: one unconstrained and one constrained (that takes into account pre - set values of Bonus coefficients).
- Table 2 compare final coefficients of both models. They have been exponentiate in order to represent relativities with respect to a base premium.
- The intercept of the constrained model, multiplied by a balancing value to totalized the expected claim cost, is the base premium that the reference insured (living in Zone 4, having bonus 0, less than 1000Km, driving a Make 9 car) will pay.

Rate relativities coefficients

levels	free	restricted
(Intercept)	445.69	435.68
Kilometres2: 1001-15000	1.24	1.23
Kilometres3: 15001-20000	1.38	1.38
Kilometres4: 20000-25 000	1.51	1.52
Kilometres5: 25001+	1.79	1.80
Make1	1.14	1.16
Make2	1.21	1.24
Make3	1.01	1.04
Make4	0.49	0.47
Make5	1.22	1.24
Make6	0.78	0.78
Make7	0.97	0.99
Make8	1.42	1.46
Zone1: Stockholm, Goteborg, Malmo	1.56	1.54
Zone2: Other large cities	1.26	1.25
Zone3: Smaller cities southern Sweden	1.12	1.11
Zone5: Smaller cities northern Sweden	1.20	1.19
Zone6: Rural areas northern Sweden	1.08	1.08
Zone7: Gotland	0.84	0.84

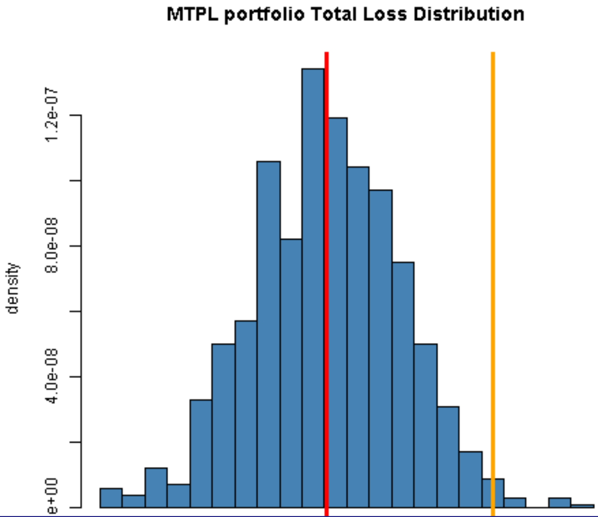
Table : MTPL tariff relativities

Capital consideration

- It is possible to generate the distribution of portfolio total losses as sum of each row simulated total losses.
- Each row total loss is the convolution of Poisson distributed claim with $\tilde{\lambda}$ equal to fitted frequency. Each claim is assumed to follow a Gamma distribution whose parameters can be calculated backward thanks to GLM theory assumptions (Equation 3).

$$\begin{cases} E\left[\tilde{c}\right] = \mu_i \\ \mathrm{var}\left[\tilde{c}\right] = \phi {\mu^2}_i \end{cases} \tag{3}$$

Capital consideration



Finalizing the tariff

- Each row total losses are simulated many times and the distribution of Portfolio total losses, $\tilde{L}$ is therefore derived, as Figure 4 shows.
- Keeping the assumption the portfolio not to change in terms of size and underwriting mix, it can be derived an average cost of capital per unit of exposures.
- Table 3 summarizes key figures regarding the MTPL pricing exercise.

Finalizing the tariff

exposures	2379212.08
frequency	0.05
severity	4955.25
average premium	243.48

Table : Personal Lines Example Key Figures

Outline

Introduction

- XL reinsurance contracts cover primary insurer for large losses. They are used to stabilize portfolios' loss experience and to reduce their volatility.
- Clark's paper [Clark, 1996] provides a very good introduction about pricing reisurance contracts.
- Typical tools used for reinsurance pricing are: loss distribution modelling, extreme value theory and convolution theory.

The XL example

- A primary insurer seeks coverage for an MTPL portfolio. Estimated exposures are 500K and the subject premium (GWP) is 250Mln.
- It asks a quote for 10xs2 Mln.
- It has provided historical ground up losses over 500K that have been trended and developed to ultimate for the analysis.

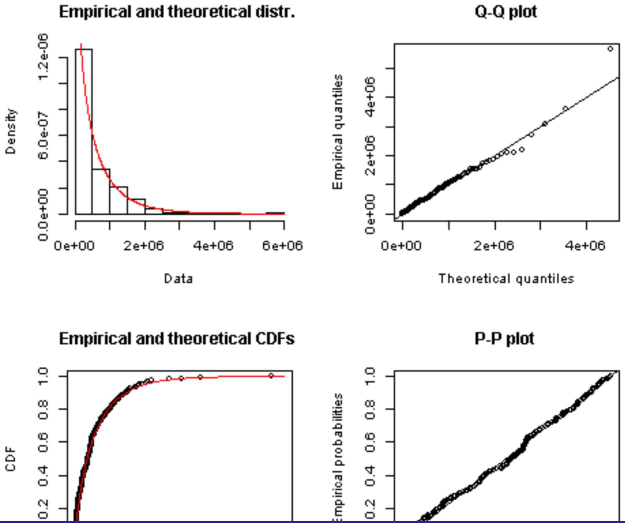
The pricing methodology

- Model the cost, $\tilde{X}$, and the frequency, $\tilde{N}$, of gross claims using distribution fitting
- Apply convolution theory to determine the total gross $\tilde{L}_{\text{gross}}$ and ceded, $\tilde{L}_{\text{ceded}}$, losses.
- Apply loads (expenses and profit contingency) and quote the price as a percentage of subject premium.

Cost modelling

- The distribution of large losses should be carefully assessed.
- *actuar* package, [Dutang et al., 2008], package provide a wide set of loss distribution densities and utility functions. R packages *fitdistrplus*, [Delignette-Muller et al., 2012], for distribution fitting, and *ADGofTestR*, [Bellosta, 2011], for AD test of goodness of fit help for it.
- The analysis on individual large losses data shows that Weibull distribution provides the better fit, as Figure 5 shows. This is well confirmed also by Anderson Darling test.

Cost Modelling



Frequency modelling

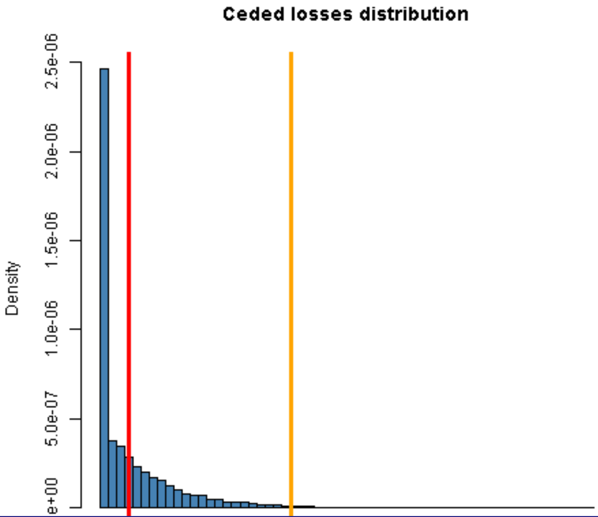
- Historical portfolio analysis (omitted) leads to set the frequency of losses in exess of 1Mln to 0.00001.
- The is need to simulate gross and ceded total losses.
- Convolution theory is used for this purpose.

Ceded loss distribution

- Gross total losses will be modelled by convolution as well as ceded ones.
- We assume the number of claim to follow a Poisson law of parameter $\lambda = E * 0.00001$.
- Each gross loss, $\tilde{X}_{gross}$, will follow Weibull distribution using parameters fitted in previous step, for the exceeding 1Mln portion.
- Ceded claims will be $\tilde{X}_{ceded} = min(L, max(0, \tilde{X} - P))$, being $L = 10Mln$ and $P = 2Mln$. Their distribution is shown in Figure 6.

$$\tilde{L} = \sum_{i=0 \ldots \tilde{N}} X_i \tag{4}$$

Ceded loss distribution



Final XL quote

subject premium (SP)	250.00
expected ceded losses	0.71
rate (percent of SP)	0.55

Table : XL Reinsurance example key figures

Outline

Life insurance code I

```
> source("lifeInsurance.R",echo=TRUE)

> library(lifecontingencies) #load the library

> data(soaLt) #load the data frame containing x and lx

> portfolioSize=1000 #define the size of the portfolio

> #price the contract
>
> actuarialtable4Example=new("actuarialtable", x=soaLt$x, lx=soaLt$Ix, interest

> oneContractDistribution=100000*rLifeContingencies(n=50000,lifecontingency="Ax

> meanValue=portfolioSize*100000*Axn(actuarialtable4Example,25,40) #mean value

> allocatedCapital=qnorm(p=0.99,mean=meanValue, sd=sd(oneContractDistribution)*

> profitLoadPerContract=0.15*allocatedCapital/portfolioSize #allocate profit to
```

Life insurance code II

```
> #final premium
> PStochastic=(100000*Axn(actuarialtable4Example,25,40)+profitLoadPerContract)/

> #save results
> png("./figures/lifeIns.png")

> hist(oneContractDistribution, col="steelblue",
+      main="Life insurance contracts benefit present value distribution")

> abline(v=mean(oneContractDistribution), col="red",lwd=3)

> abline(v=quantile(oneContractDistribution, p=0.99), col="orange",lwd=3)

> dev.off()
null device
          1

> infos=c("size", "capital", "duration", "age", "expense ratio", "final premium

> value=c(1000, 100000, 40, 25, 0.03, PStochastic) #save data
```

Life insurance code III

```
> dfInfoLife=data.frame(infos, value)

> #out to latex
>
> print(xtable(dfInfoLife, caption="Life insurance example key figures",
+              label="tab:lifeIns"), include.row=FALSE, i .... [TRUNCATED]
```

MTPL pricing code I

```
> source("personalLines.R",echo=TRUE)

> #load data
> data(motorins,package="faraway")

> #data preparation using descriptions in motorins
> dataset<-transform(motorins,
+                    Kilometres=factor(Kilometres, levels=c(1:5),
+ .... [TRUNCATED]

> #set the reference level the one with most exposure
> dataset$Zone=relevel(dataset$Zone, ref="4: Rural areas southern Sweden")

> dataset$Make=relevel(dataset$Make, ref="9")

> #define the bonus malus scale
>   bonusCoefficient<-function(varBonus)
+   {
+     out=NULL
+     if(varBonus==1) out=1 else
```

MTPL pricing code II

```
+       if(varBonus==2 .... [TRUNCATED]

> attach(dataset)
The following object(s) are masked from 'dataset (position 3)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'dataset (position 4)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'dataset (position 5)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'dataset (position 6)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'dataset (position 7)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'dataset (position 8)':
```

MTPL pricing code III

```
    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'temp (position 9)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'temp (position 10)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'temp (position 11)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone
The following object(s) are masked from 'dataset (position 12)':

    Bonus, BonusNum, Claims, Insured, Kilometres, Make, Payment, perd, Zone

> ncdCoeff=sapply(as.numeric(Bonus),bonusCoefficient)

> detach(dataset)

> dataset$ncdCoeff=ncdCoeff
```

MTPL pricing code IV

```
> freqModel<-glm(Claims~Kilometres+Zone+Bonus+
+                  Make,offset=log(Insured), data=dataset,family="poisson")

> library(coefplot)

> png("./figures/modelFreq.png")

> coefplot(model=freqModel, intercept=FALSE, title="Frequency model coef plot",

> dev.off()
null device
          1

> #severity modelling
>
> 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 ..." ... [
```

MTPL pricing code V

```
> 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
```

MTPL pricing code VI

```
> #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
```

MTPL pricing code VII

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

> png("./figures/sevModel.png")

> coefplot(model=sevModel, intercept=FALSE, title="Severity model coef plot", y

> dev.off()
null device
          1

> #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,
```

MTPL pricing code VIII

```
+                     fittedFrequency=fittedFrequency,
+              .... [TRUNCATED]

> #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,
+                ....  [TRUNCATED]

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

> ########################################
> #model free and constrained pure premiums
```

MTPL pricing code IX

```
> ########################################
> burnCostFreeModel< .... [TRUNCATED]

> #this is the burning cost model with bonus malus coefficients constrained
> burnCostConstrModel<-glm(purePremium~Zone+Make+Kilometres,
+           .... [TRUNCATED]

> #add the risk premium to the dataset
> dataset$risk_premium=predict(burnCostConstrModel,
+                              newdata=dataset, type="resp ..." ... [TRUNCATED]

> #check the balance
> #actual total losses
> balancingValue<-with(dataset, sum(Payment)/sum(Insured*risk_premium))

> #5% delta
>
> ###################################################
> ### coefficient comparisons
> ################################################# .... [TRUNCATED]
```

MTPL pricing code X

```
> coefConstr<-coef(burnCostConstrModel)

> #create a pretty table and exponentiate coefficients
> tableCoefFree<-data.frame(levels=names(coefFree),
+                           coefficientsFr .... [TRUNCATED]

> tableCoefConstr<-data.frame(levels=names(coefConstr),
+                             coefficientRestricted=exp(as.numeric(coefConstr))

> tableCoeff<-merge(x=tableCoefFree,y=tableCoefConstr)

> names(tableCoeff)[2:3]<-c("free","restricted")

> print(xtable(tableCoeff, caption="MTPL tariff relativities", label="tab:MTPLR
+       include.row=FALSE, include.col=TRUE, size="scriptsize",
 .... [TRUNCATED]

> #calculate the variance
> temp<-dataset
```

MTPL pricing code XI

```
> temp$costDisp=2.983458 #fDisprom Severity Model

> temp$costVar=with(temp, averageCost^2*costDisp) #calculate the cost condition

> temp$scaleCost=with(temp, costVar/averageCost) #calculate the scale

> temp$shapeCost=with(temp, averageCost/scaleCost)

> dataset<-temp

> #
> #nsim=1000
> #simMatrix<-matrix(0, nrow=nrow(dataset),ncol=nsim)
> # .convolve<-function(num, shape, scale)
> # {
> #   out=numeric(1)
> #   ou .... [TRUNCATED]

> #finalizing the tariff
```

MTPL pricing code XII

```
>
> #cost of capital
>
> MTPLCoC=0.15*(quantile(portfolioLosses,.99)-mean(portfolioLosses))

> MTPLCoCxInsured=MTPLCoC/sum(dataset$Insured)

> dataset$FinalCommercialPremium=(MTPLCoCxInsured+balancingValue*dataset$risk_p

> png("./figures/mtplPTFdist.png")

> hist(portfolioLosses, main="MTPL portfolio Total Loss Distribution", xlab="to

> abline(v=mean(portfolioLosses), col="red",lwd=3)

> abline(v=quantile(portfolioLosses, p=0.99),col="orange",lwd=3)

> dev.off()
null device
          1
```

MTPL pricing code XIII

```
> #summarize
> attach(dataset)
The following object(s) are masked _by_ '.GlobalEnv':

    fittedFrequency, fittedSeverity, ncdCoeff
The following object(s) are masked from 'dataset (position 3)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone
The following object(s) are masked from 'dataset (position 4)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone
The following object(s) are masked from 'dataset (position 5)':
```

MTPL pricing code XIV

```
    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone
The following object(s) are masked from 'dataset (position 6)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone
The following object(s) are masked from 'dataset (position 7)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone
The following object(s) are masked from 'dataset (position 8)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
```

MTPL pricing code XV

```
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone
The following object(s) are masked from 'temp (position 9)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar, fittedFrequency,
    fittedSeverity, Insured, Kilometres, Make, ncdCoeff, Payment, perd,
    purePremium, risk_premium, scaleCost, shapeCost, Zone
The following object(s) are masked from 'temp (position 10)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar, fittedFrequency,
    fittedSeverity, Insured, Kilometres, Make, ncdCoeff, Payment, perd,
    purePremium, risk_premium, scaleCost, shapeCost, Zone
The following object(s) are masked from 'temp (position 11)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar, fittedFrequency,
    fittedSeverity, Insured, Kilometres, Make, ncdCoeff, Payment, perd,
    purePremium, risk_premium, scaleCost, shapeCost, Zone
The following object(s) are masked from 'dataset (position 12)':

    averageCost, Bonus, BonusNum, Claims, costDisp, costVar,
```

MTPL pricing code XVI

```
    FinalCommercialPremium, fittedFrequency, fittedSeverity, Insured,
    Kilometres, Make, ncdCoeff, Payment, perd, purePremium, risk_premium,
    scaleCost, shapeCost, Zone

> infos=c("exposures", "frequency", "severity", "average premium")

> values=c(sum(Insured), sum(Claims)/sum(Insured), sum(Payment)/sum(Claims),
+          sum(FinalCommercialPremium*Insured)/sum(Insured))

> dfInfoMTPL=data.frame(infos, values)

> print(xtable(dfInfoMTPL, caption="Personal Lines Example Key Figures",
+              label="tab:personalLinesSummary"),
+       include.row=FALSE .... [TRUNCATED]
```

XL Code I

```
> source("xlpricing.R",echo=TRUE)

> #pricing xl contracts with R
>
> #severity fitting
>
> library(fitdistrplus) #load fitdistrplus package

> library(actuar) #load package for distributions

> library(ADGofTest) #load package to fit anderson darling

> load(file="xlClaims.RData")

> claims2analyze=mtplLargeClaims$ultimate_cost[mtplLargeClaims$ultimate_cost>10

> #try various function
> fitGamma<-fitdist(data=claims2analyze, distr="gamma", method="mme") #gamma wi

> fitWeibull<-fitdist(data=claims2analyze, distr="weibull", method="mle") #Weib
```

XL Code II

```
>  #lognormal
> fitLognormal<-fitdist(data=claims2analyze, distr="lnorm", method="mge",gof="A

> #fitçLogistic<-fitdist(data=claims2analyze, distr="logis", method="mle")
>
> #check wich bet
>
> #check which distribution is better
> ad.test(cla .... [TRUNCATED]

Anderson-Darling GoF Test

data:  claims2analyze  and  pgamma
AD = 0.4421, p-value = 0.806
alternative hypothesis: NA


> ad.test(claims2analyze, pweibull, fitWeibull$estimate[1],fitWeibull$estimate[
```

XL Code III

```
Anderson-Darling GoF Test

data:  claims2analyze  and  pweibull
AD = 0.2758, p-value = 0.9551
alternative hypothesis: NA


> ad.test(claims2analyze, plnorm, fitLognormal$estimate[1],fitLognormal$estimat

Anderson-Darling GoF Test

data:  claims2analyze  and  plnorm
AD = 4.2878, p-value = 0.006332
alternative hypothesis: NA


> png("./figures/weibullFit.png")

> plot(fitWeibull)
```

XL Code IV

```
> dev.off()
null device
          1

> #apply convolution to find the ceded total loss distribution
>
> E=500000

> freq=1*10^-5

> lambda=E*freq

> nsim=10000

> grossLosses=numeric(nsim)

> cededLosses=numeric(nsim)

> netLosses=numeric(nsim)
```

XL Code V

```
> #function to get ceded losses
> .getCeded<-function(grossLoss, priority, limit)
+ {
+   out=numeric(1)
+   out=min(limit, max(0, grossLoss-priority) .... [TRUNCATED]

> #convolution modelling
>
> for(i in 1:nsim)
+ {
+   num=rpois(1, lambda) #simulate the number of claims
+   if(num==0) {
+     grossLosses[i]=0
+   .... [TRUNCATED]

> #price the contract
>
> expRatio=0.05

> SubjectPremium=250*10^6
```

XL Code VI

```
> allocatedCapital=quantile(cededLosses,.99)-mean(cededLosses)

> burningCost=(mean(cededLosses)+0.15*allocatedCapital)/(1-expRatio)

> rate=(burningCost)/SubjectPremium

> infos=c("subject premium (SP)", "expected ceded losses", "rate (percent of SP

> value=c(SubjectPremium/10^6, mean(cededLosses)*10^-6, rate*100 ) #save data

> dfInfoXL=data.frame(infos, value)

> #out to latex
> png("./figures/xlCededDistr.png")

> hist(cededLosses, breaks=50, col="steelblue", main="Ceded losses distribution

> dev.off()
null device
```

XL Code VII

1

```
> print(xtable(dfInfoXL, caption="XL Reinsurance example key figures", digit=2,
+              label="tab:xlReinsurance"), include.row=FALSE, include. .... [TR
```

Reference I

Bellosta, C. J. G. (2011).
ADGofTest: Anderson-Darling GoF test.
R package version 0.3.

Casualty Actuarial Society (1988).
Statement of principles regarding property and casualty insurance ratemaking.
http://www.casact.org/professionalism/standards/princip/sppcrate.pdf.
Online; accessed 25-05-2013.

Clark, D. R. (1996).
Basics of reinsurance pricing.
CAS Study Note, pages 41–43.

Delignette-Muller, M. L., Pouillot, R., Denis, J.-B., and Dutang, C. (2012).
fitdistrplus: help to fit of a parametric distribution to non-censored or censored data.
R package version 1.0-0.

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

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

Reference II

IBM Corp (2012).
SPSS, Release 21.0 Advanced Statistical Procedures Companion.
IBM Corp., Armonk, NY.

Lander, J. P. (2013).
coefplot: Plots Coefficients from Fitted Models.
R package version 1.2.0.

R Development Core Team (2012).
R: A Language and Environment for Statistical Computing.
R Foundation for Statistical Computing, Vienna, Austria.
ISBN 3-900051-07-0.

SAS Institute Inc. (2011).
SAS/STAT Software, Version9.3.
Cary, NC.

Spedicato, G. A. (2012).
Third party motor liability ratemaking with r.
`http://www.casact.org/research/wp/papers/working-paper-spedicato-%202012-06.pdf`.
Casualty Actuarial Society Working Paper.

Spedicato, G. A. (2013).
Lifecontingencies: an R package to perform life contingencies actuarial mathematics.
R package version 0.9.7.

Reference III

The MathWorks Inc. (2010).
MATLAB Version 7.10.0 (R2010a).
The MathWorks Inc., Natick, Massachusetts.

Zhang, W. (2012).
cplm: Compound Poisson linear models.
R package version 0.6-4.