Other Liability Insurance (Mercury Insurance Group)

Jenny Liang (Wen Liang)
Email: wliang88@berkeley.edu

```
###### function to read data
OthLiab=read.csv("/Users/JennyLiang/Desktop/OthLiab_pos.csv",header=TRUE)
grp.code=unique(OthLiab$GRCODE)
ins.line.data=function(g.code){
 b=subset(OthLiab,OthLiab$GRCODE==g.code)
 name=b$GRNAME
 grpcode=b$GRCODE
 ay=b$AccidentYear
 dev=b$DevelopmentLag
 cum_pdloss=b[,7]
 cum_incloss=b[,6]
 data.out=data.frame(name,grpcode,ay,dev,cum_pdloss,cum_incloss)
 return(data.out)
}

###### my data
which(ins.line.data(grp.code)[,1]=="First Mercury Ins Co")   ##[1] 64
data=ins.line.data(grp.code[64])

###### upper triangle
u_triangle=subset(data,ay+dev<=1998)

###### install necessary packages
 ##install.packages("lme4")
 ##install.packages("geepack")
 ##install.packages("ChainLadder")
 ##install.packages("MuMIn")
 ##install.packages("MESS")
library(lme4)
library(geepack)
library(ChainLadder)
library(MuMIn)
library(MESS)

##########################data in triangles
triangle=as.triangle(u_triangle,origin="ay", dev="dev",
                value="cum_pdloss")
###### plot the data
triangle
plot(triangle, xlab="Development Year",
   ylab="Cumulative Claims",lwd="5",font.axis = 4, cex.axis = 1.2,
```

```
    cex.lab = 1.5, col.lab = "dark red" )
plot(triangle,lattice=T)

########################################ChainLadder Model
###incremental data, function was written before knowing the cum2incr function
inc=u_triangle[5]
inc
for (k in (2:91)){
  if ( is.na(u_triangle[k,3])!=TRUE & u_triangle[k,3]==u_triangle[k-1,3])
    inc[k,1]=u_triangle[k,5]-u_triangle[k-1,5]
  else if (is.na(u_triangle[k,3])!=TRUE) inc[k,1]=u_triangle[k,5]
}
u_triangle[5]=inc

inc_data=cbind(u_triangle[3],u_triangle[4],u_triangle[5])
colnames(inc_data)[3]="inc_loss"

###### incremental data
inc_triangle=cum2incr(triangle)
inc_triangle[1,]  ##Show first origin period and its incremental developement
inc_triangle
plot(inc_triangle, xlab="Development Year",
    ylab="Cumulative Claims",lwd="5",font.axis = 4, cex.axis = 1.2,
    cex.lab = 1.5, col.lab = "dark red" )
plot(inc_triangle,lattice=T)

###### Basic Idea
## Link ratios are calculated as the volumne weighted average developement
## rations of a cumulative loss developement triangle from one development period
## to the next

year=10
f=sapply(1:(year-1),
       function(i){
         sum(triangle[c(1:(year-i)),i+1])/sum(triangle[c(1:(year-i)),i])
       })
f ##[1] 5.592638 2.858319 1.624835 1.322425 1.174066 1.076317 1.020656
1.011057 1.049548

## The oldest origin year is not fully developed, approach to extrapolate
## the developement raions, assuming a log-linear model

dev.lag=1:(year-1)
plot(log(f-1)~dev.lag, main="Log-linear extrapolation of age-to-age factors")
NewModel=lm(log(f-1)~dev.lag)
abline(NewModel)
```

```
co=coef(NewModel)

###### extrapolate another 100 development period
tail=exp(co[1]+c((year+1):(year+100))*co[2])+1
f.tail=prod(tail)
f.tail##[1] 1.005696

###### plot the expected claims development patterns
plot(100*(rev(1/cumprod(rev(c(f,tail[tail>0.999])))))),t="b",
    main="Expected claims development pattern",
    xlim=c(0,15),
    xlab="Development Period", ylab="Development % of ultimate loss")

##### to forecast the next developemtn period
f=c(f,f.tail)
full_triangle=cbind(triangle,Ult=rep(0,10))
for (k in 1:year){
  full_triangle[(year-k+1):year,k+1]=full_triangle[(year-k+1):year,k]*f[k]
}
full_triangle=round(full_triangle)
full_triangle
sum(full_triangle[,11]-getLatestCumulative(triangle)) ##[1] 29259

##### Mack chain-ladder
mack=MackChainLadder(triangle, est.sigma="Mack")
mack$f
round(mack$FullTriangle)
plot(mack)
plot(mack,lattice=T)
```