Provisionnement

EL KAMLI Adam

2024-10-06

Importation des données

```
library(readxl)
library(ChainLadder)
```

```
## Warning: le package 'ChainLadder' a été compilé avec la version R 4.3.3
```

```
##
## Welcome to ChainLadder version 0.2.18
##
##
## Pour citer le package 'ChainLadder' dans une publication, utilisez :
##
##   Gesmann M, Murphy D, Zhang Y, Carrato A, Wuthrich M, Concina F, Dal
##   Moro E (2023). _ChainLadder: Statistical Methods and Models for
##   Claims Reserving in General Insurance_. R package version 0.2.18,
##   <https://CRAN.R-project.org/package=ChainLadder>.
##
## Une entrée BibTeX pour les utilisateurs LaTeX est
##
##   @Manual{,
##     title = {ChainLadder: Statistical Methods and Models for Claims Reserving in General
## Insurance},
##     author = {Markus Gesmann and Daniel Murphy and Yanwei (Wayne) Zhang and Alessandro Carrato and Ma
##     year = {2023},
##     note = {R package version 0.2.18},
##     url = {https://CRAN.R-project.org/package=ChainLadder},
##   }
##
## To suppress this message use:
## suppressPackageStartupMessages(library(ChainLadder))
```

```
library(tidyverse)
```

```
## Warning: le package 'lubridate' a été compilé avec la version R 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
Prov <- read_excel("C:\\Users\\user\\OneDrive\\Bureau\\S4\\S4-P2\\Assurance non vie\\provisions.xlsx")
Prov <- as.data.frame(Prov)
```

On calcule les pertes cumules

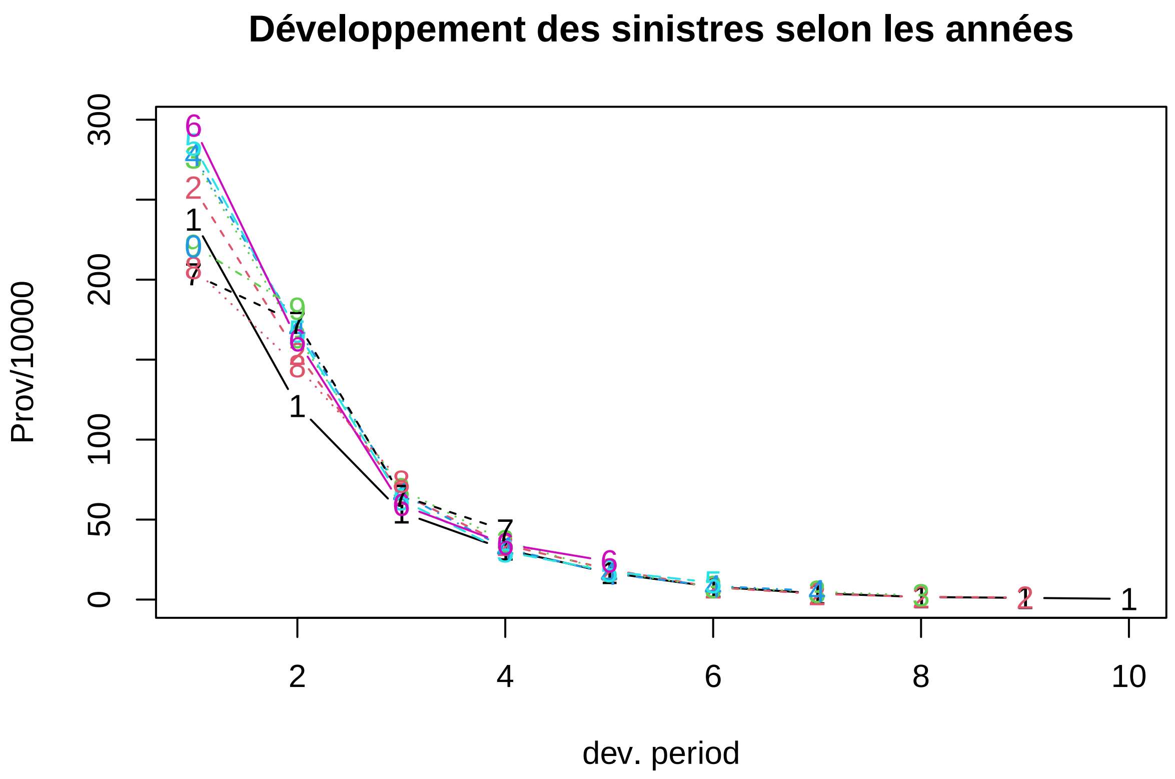
```
rownames(Prov) <-  Prov[,1]
Prov = Prov[2:length(Prov[1,])]
head(Prov)
```

```
##            0       1      2      3      4      5     6     7     8    9
## 2014 2376384 1211168 545883 313790 167151  80072 39235 16030 10664 4256
## 2015 2576278 1537150 662445 342694 188799  77047 35042 17199 13413   NA
## 2016 2763277 1640231 688959 364199 177108  78169 48371 26377    NA   NA
## 2017 2779698 1698531 661401 321434 162578  84581 54450    NA    NA   NA
## 2018 2843224 1673604 624401 299473 176842 106296    NA    NA    NA   NA
## 2019 2962385 1620298 591932 347434 238375     NA    NA    NA    NA   NA
```

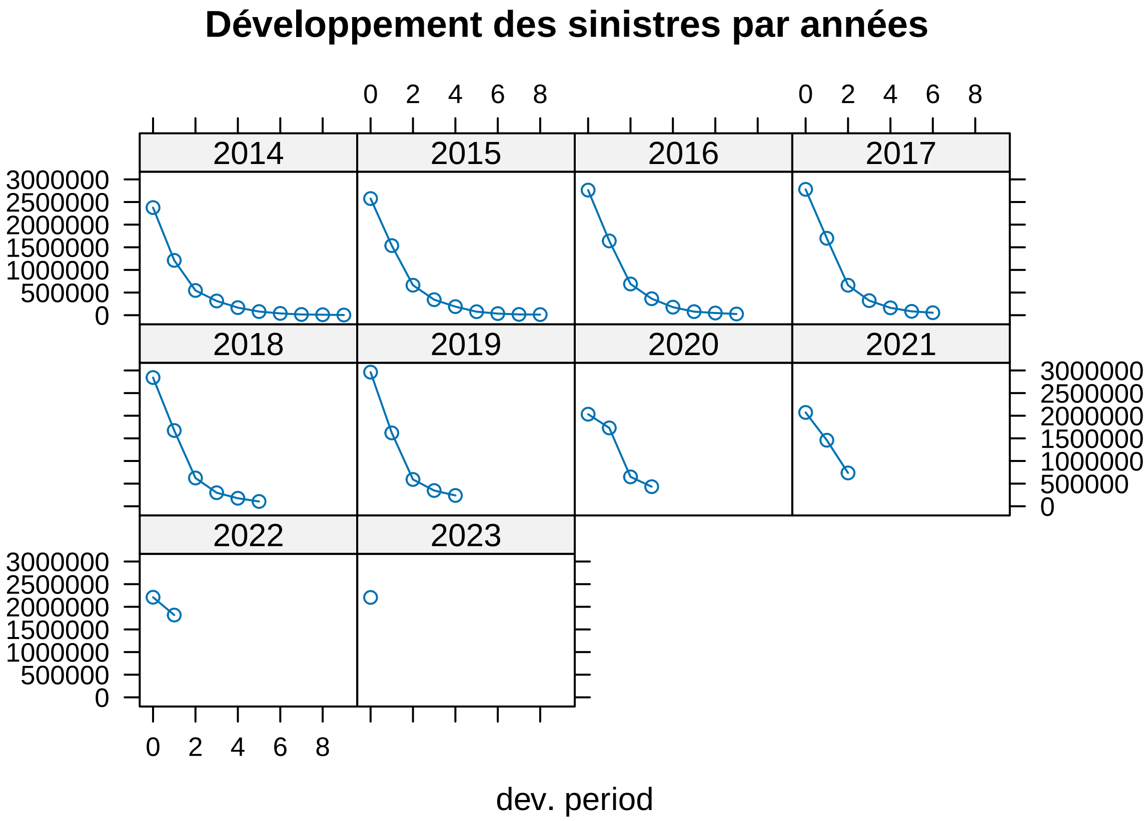
```
Prov <- as.triangle(as.matrix(Prov))
head(Prov)
```

```
##       dev
## origin       0       1      2      3      4      5     6     7     8    9
##   2014 2376384 1211168 545883 313790 167151  80072 39235 16030 10664 4256
##   2015 2576278 1537150 662445 342694 188799  77047 35042 17199 13413   NA
##   2016 2763277 1640231 688959 364199 177108  78169 48371 26377    NA   NA
##   2017 2779698 1698531 661401 321434 162578  84581 54450    NA    NA   NA
##   2018 2843224 1673604 624401 299473 176842 106296    NA    NA    NA   NA
##   2019 2962385 1620298 591932 347434 238375     NA    NA    NA    NA   NA
```

```
plot(Prov/10000, main = "Développement des sinistres selon les années")
```



```
plot(Prov, lattice= T, main = "Développement des sinistres par années")
```



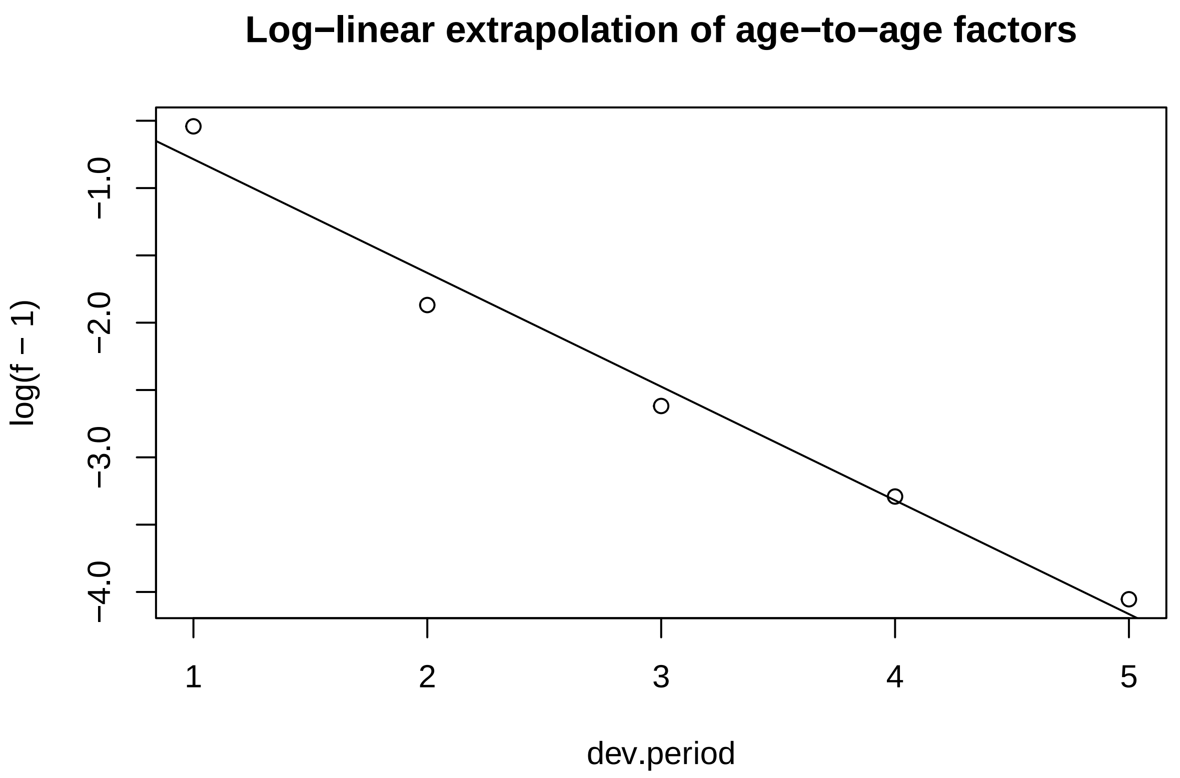
```
Prov_cum <- incr2cum(Prov)   #hd
```

Chain ladder

```
n <- 6
f <- sapply(1:(n-1),
            function(i){
              sum(Prov_cum[c(1:(n-i)),i+1])/sum(Prov_cum[c(1:(n-i)),i])
            }
)
f
```

```
## [1] 1.581810 1.154298 1.072897 1.037211 1.017353
```

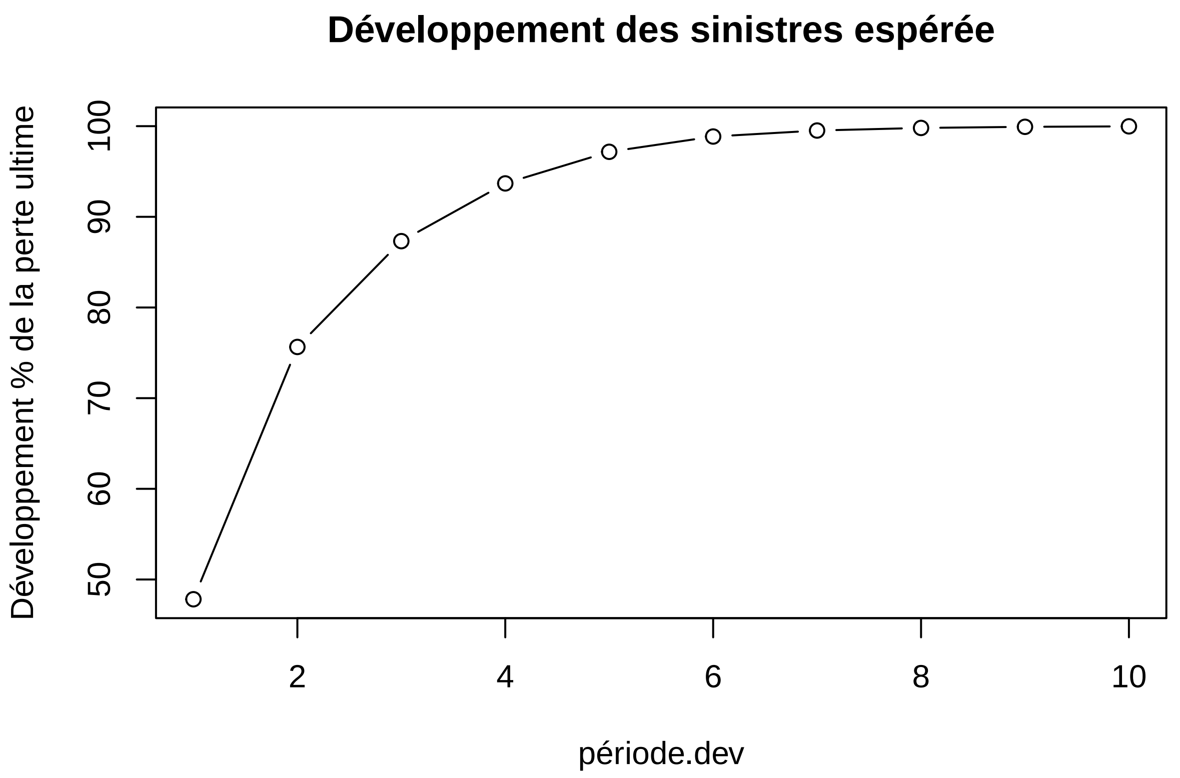
```
dev.period <- 1:(n-1)
plot(log(f-1) ~ dev.period,
     main="Log-linear extrapolation of age-to-age factors")
tail.model <- lm(log(f-1) ~ dev.period)
abline(tail.model)
```



```
co <- coef(tail.model)
tail <- exp(co[1] + c(n:(n + 100)) * co[2]) + 1
f.tail <- prod(tail)
f.tail
```

```
## [1] 1.01175
```

```
plot(100*(rev(1/cumprod(rev(c(f, tail[tail>1.0001]))))), t="b",
     main="Développement des sinistres espérée",
     xlab="période.dev", ylab="Développement % de la perte ultime")
```



```
f <- c(f, f.tail)
full_prov <- cbind(Prov_cum, Ult = rep(0, 6))
```

```
## Warning in cbind(Prov_cum, Ult = rep(0, 6)): number of rows of result is not a
## multiple of vector length (arg 2)
```

```
for(k in 1:n){
  full_prov[(n-k+1):n, k+1] <- full_prov[(n-k+1):n,k]*f[k]
}
round(full_prov)
```

```
##            0       1       2       3       4       5       6       7       8
## 2014 2376384 3587552 4133435 4447225 4614376 4694448 4749607 4749713 4760377
## 2015 2576278 4113428 4775873 5118567 5307366 5399463 5462906 5436654 5450067
## 2016 2763277 4403508 5092467 5456666 5659712 5757924 5825579 5786691      NA
## 2017 2779698 4478229 5139630 5514292 5719483 5818731 5887101      NA      NA
## 2018 2843224 4516828 5213768 5593834 5801985 5902665 5972020      NA      NA
## 2019 2962385 4685931 5408962 5803258 6019201 6123651 6195603      NA      NA
## 2020 2033371 3763913 4412813 4845061      NA      NA      NA      NA      NA
## 2021 2072061 3530602 4266700      NA      NA      NA      NA      NA      NA
## 2022 2210754 4028255      NA      NA      NA      NA      NA      NA      NA
## 2023 2206886      NA      NA      NA      NA      NA      NA      NA      NA
##            9 Ult
## 2014 4764633   0
## 2015      NA   0
## 2016      NA   0
## 2017      NA   0
## 2018      NA   0
## 2019      NA   0
## 2020      NA   0
## 2021      NA   0
## 2022      NA   0
## 2023      NA   0
```

```
reserve <- function(Prov_cum){
 tt = as.matrix(Prov_cum)
 chargeultime = Prov_cum[,length(Prov_cum[,1])]
 paiements= diag(tt[,6:1])
 r = chargeultime-paiements
 return(r)
}
reserve(Prov_cum)
```

```
## Warning in chargeultime - paiements: la taille d’un objet plus long n’est pas
## multiple de la taille d’un objet plus court
```

```
##  2014  2015  2016  2017  2018  2019  2020  2021  2022  2023
## 70185    NA    NA    NA    NA    NA    NA    NA    NA    NA
```

MAck

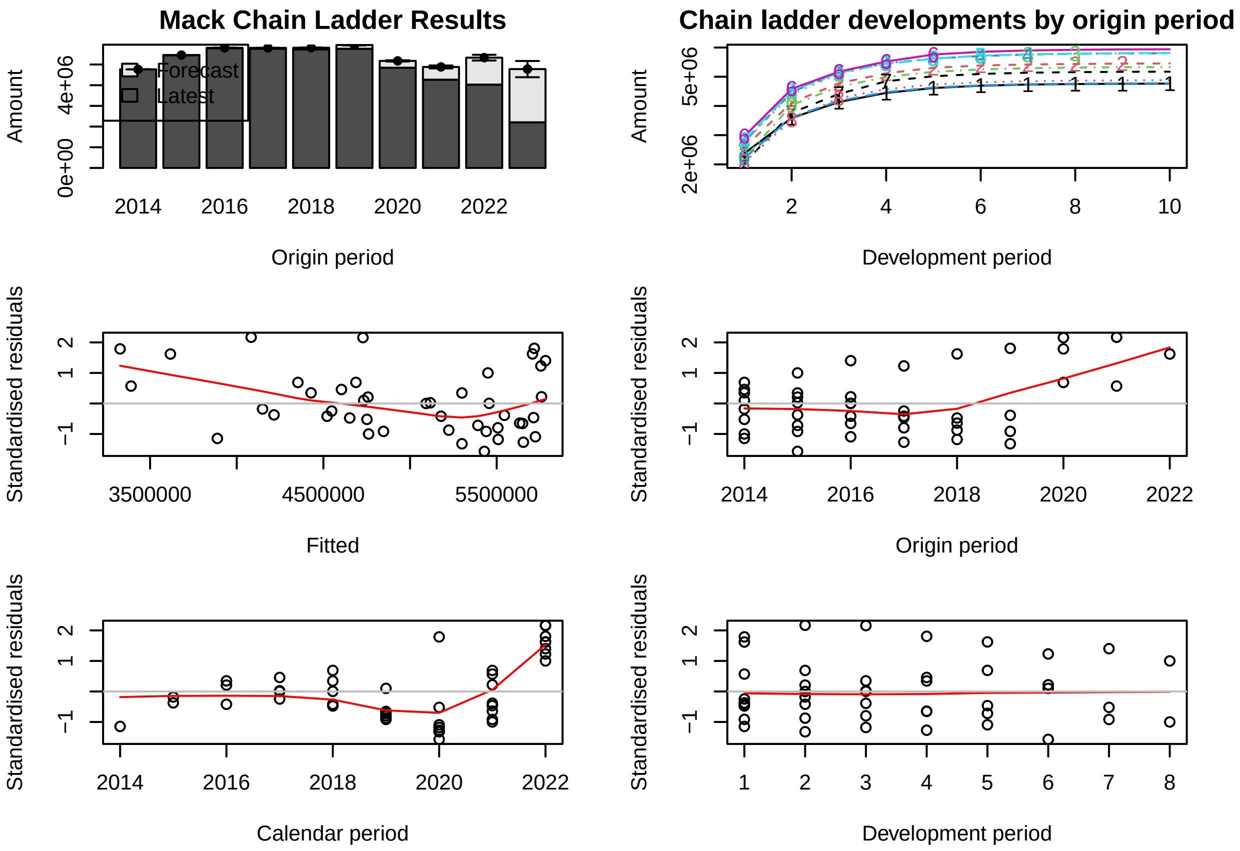
```
mack <- MackChainLadder(Prov_cum, est.sigma="Mack")
mack
```

```
## MackChainLadder(Triangle = Prov_cum, est.sigma = "Mack")
##
##         Latest Dev.To.Date  Ultimate      IBNR Mack.S.E CV(IBNR)
## 2014 4,764,633       1.000 4,764,633         0        0      NaN
## 2015 5,450,067       0.999 5,454,940     4,873      239   0.0490
## 2016 5,786,691       0.997 5,805,555    18,864    1,095   0.0580
## 2017 5,762,673       0.993 5,803,113    40,440    5,140   0.1271
## 2018 5,723,840       0.985 5,811,489    87,649    9,514   0.1086
## 2019 5,760,424       0.969 5,941,648   181,224   16,450   0.0908
## 2020 4,845,061       0.936 5,174,026   328,965   31,322   0.0952
## 2021 4,266,700       0.874 4,882,119   615,419   71,432   0.1161
## 2022 4,028,255       0.756 5,330,516 1,302,261  138,183   0.1061
## 2023 2,206,886       0.462 4,778,035 2,571,149  392,504   0.1527
##
##                   Totals
## Latest:   48,595,230.00
## Dev:               0.90
## Ultimate: 53,746,072.86
## IBNR:      5,150,842.86
## Mack.S.E     432,923.14
## CV(IBNR):          0.08
```

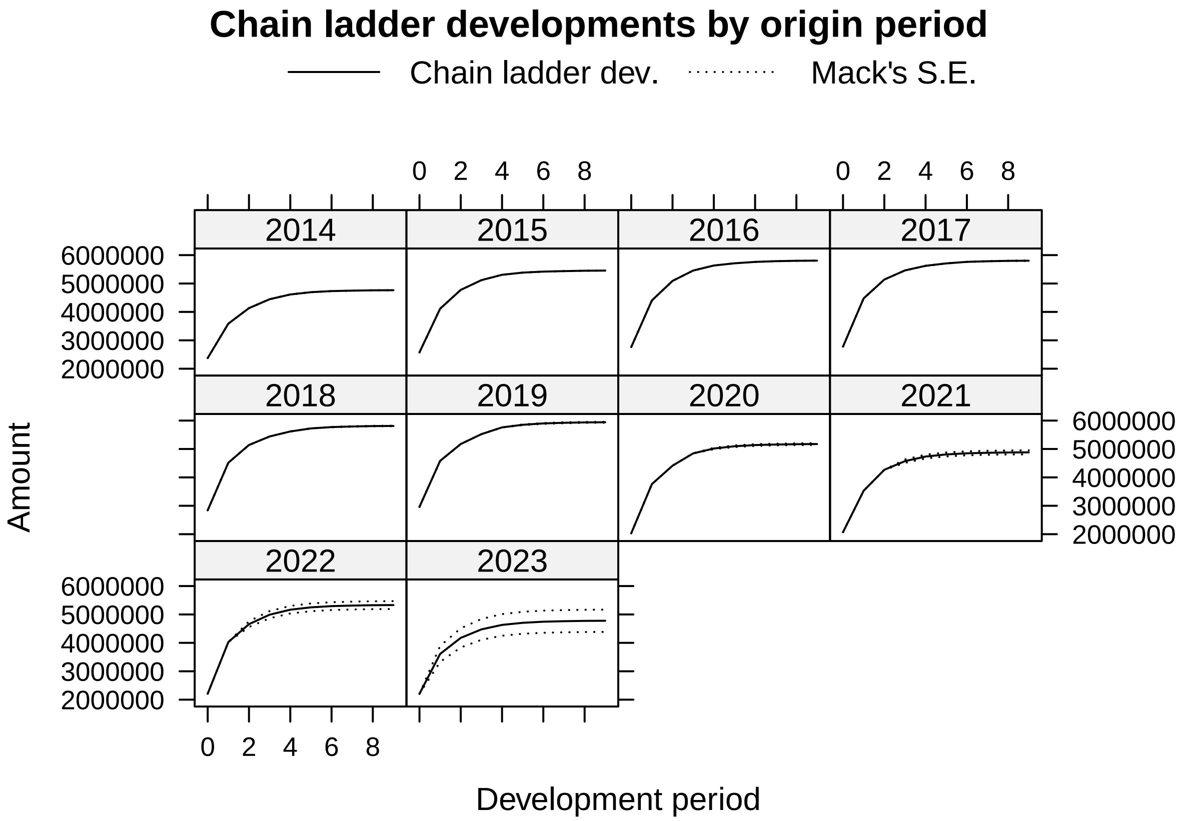
```
mack$FullTriangle
```

```
##       dev
## origin       0       1       2       3       4       5       6       7       8
##   2014 2376384 3587552 4133435 4447225 4614376 4694448 4733683 4749713 4760377
##   2015 2576278 4113428 4775873 5118567 5307366 5384413 5419455 5436654 5450067
##   2016 2763277 4403508 5092467 5456666 5633774 5711943 5760314 5786691 5800369
##   2017 2779698 4478229 5139630 5461064 5623642 5708223 5762673 5784258 5797930
##   2018 2843224 4516828 5141229 5440702 5617544 5723840 5770990 5792606 5806298
##   2019 2962385 4582683 5174615 5522049 5760424 5852036 5900242 5922342 5936340
##   2020 2033371 3763913 4412813 4845061 5016215 5095991 5137969 5157214 5169404
##   2021 2072061 3530602 4266700 4571714 4733212 4808487 4848097 4866256 4877758
##   2022 2210754 4028255 4658574 4991602 5167932 5250121 5293369 5313196 5325755
##   2023 2206886 3610746 4175736 4474247 4632302 4705972 4744738 4762510 4773767
##       dev
## origin       9
##   2014 4764633
##   2015 5454940
##   2016 5805555
##   2017 5803113
##   2018 5811489
##   2019 5941648
##   2020 5174026
##   2021 4882119
##   2022 5330516
##   2023 4778035
```

```
plot(mack)
```



```
plot(mack, lattice = T)
```



GLM

```
ligne <- rep(1:n,n)
colonne <- rep(1:n, each = n)
X <- as.vector(Prov_cum)
lig <- as.factor(ligne)
col <- as.factor(colonne)
prov_glm <- as.data.frame(cbind(X, lig, col))
```

```
## Warning in cbind(X, lig, col): number of rows of result is not a multiple of
## vector length (arg 2)
```

```
fit1 <- glm(X~lig+col , data = prov_glm, family = Gamma(link = "log"))
summary(fit1)
```

```
##
## Call:
## glm(formula = X ~ lig + col, family = Gamma(link = "log"), data = prov_glm)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.061041   0.097780 154.030  < 2e-16 ***
## lig          0.008425   0.019563   0.431 0.668493
## col          0.069679   0.019779   3.523 0.000899 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.06216729)
##
##     Null deviance: 4.3364  on 54  degrees of freedom
## Residual deviance: 3.5663  on 52  degrees of freedom
##   (45 observations effacées parce que manquantes)
## AIC: 1695.5
##
## Number of Fisher Scoring iterations: 4
```