Package: DCL (via r-universe)

November 6, 2024

Type Package

Title Claims Reserving under the Double Chain Ladder Model

Version 0.1.2

Date 2022-05-05

Depends lattice, latticeExtra

Author Maria Dolores Martinez-Miranda, Jens Perch Nielsen and Richard Verrall

Maintainer Maria Dolores Martinez-Miranda <mmiranda@ugr.es>

Description Statistical modelling and forecasting in claims reserving in non-life insurance under the Double Chain Ladder framework by Martinez-Miranda, Nielsen and Verrall (2012).

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-05 16:40:02 UTC

Config/pak/sysreqs libjpeg-dev libpng-dev

Contents

`DCL-package` *Claims Reserving under the Double Chain Ladder Model*

Description

This package provides functions for statistical modelling and forecasting in claims reserving in non-life insurance under the Double Chain Ladder framework by Martinez-Miranda, Nielsen and Verrall (2012). Using specific functions, the user will be able generate plots to visualize and gain intuition about the data (run-off triangles), break down classical chain ladder under the DCL model, visualize the underlying delay function and the inflation, introduce expert knowledge about the severity inflation, the zero-claims etc. Besides a validation exercise can be performed through a back-test on the data.

Details

Package:	DCL
Type:	Package
Version:	0.1.0
Date:	2013-10-24
License:	GPL-2

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

Maintainer: Maria Dolores Martinez-Miranda <mmiranda@ugr.es>

References

Martinez-Miranda M.D., Nielsen B, Nielsen J.P and Verrall, R. (2011) Cash flow simulation for a model of outstanding liabilities based on claim amounts and claim numbers. *Astin Bulletin*, 41/1, 107-129.

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North American Actuarial Journal*, 17(2), 101-113.

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

See more at http://www.cassknowledge.com/research/article/double-chain-ladder-cass-knowledge

Examples

```
data(NtriangleDCL)
data(XtriangleDCL)

# Classical chain ladder parameters
my.clm.par<-clm(XtriangleDCL)
Plot.clm.par(my.clm.par)

# Estimation of the DCL parameters (break-down of the chain ladder parameters)
my.dcl.par<-dcl.estimation(XtriangleDCL,NtriangleDCL)
Plot.dcl.par(my.dcl.par)

# DCL Predictions by diagonals (future calendar years)
# Splitting the chain ladder reserve into RBNR and IBNR claims (ignoring the tail)
preds.dcl.diag<-dcl.predict(my.dcl.par,Model=0,Tail=FALSE,num.dec=0)

# Full cashflow considering the tail (only the variance process)
# Below only B=200 simulations for faster calculations in the example
boot1<-dcl.boot(dcl.par=my.dcl.par,Ntriangle=NtriangleDCL,boot.type=1,B=200)
Plot.cashflow(boot1)
```

`Aggregate` *Switch to a higher level of aggregation*

Description

From the input run-off triangle (weekly, monthly, quarterly, etc.) the function creates another triangle on a higher level of aggregation.

Usage

```
Aggregate(triangle, freq = 4)
```

Arguments

- `triangle` The original run-off (incremental) triangle. It should be a squared matrix (let denote by m its dimension m).
- `freq` The frequency to be considered in the aggregation. The default value is 4, to be used to construct a yearly run-off triangle from a quarterly `triangle`

.

Details

If the input triangle does not consist of complete periods (for example a quarterly triangle with only three quarters in the last year), then the last (lower level) periods will been removed to get full aggregated periods.

Value

A run-off triangle in the specified higher level of aggregation.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

See Also

`get.incremental`,`get.cumulative`,`Plot.triangle`

Examples

```
## A dummy example: a run-off triangle with 5*4=20 quarters
m<-20
my.square<-matrix(1,m,m)
# Now my.triangle is a quarterly triangle (the upper left triangle from my.square)
my.triangle<-my.square
my.triangle[row(my.square)+col(my.square)>(m+1)]<-NA
my.yearly.triangle<-Aggregate(my.triangle)
list(original=my.triangle,yearly=my.yearly.triangle)
```

---

bdcl.estimation *Parameter estimation - DCL model using the BDCL method*

Description

Estimate the parameters in the Double Chain Ladder model (delay parameters, severity mean and variance) using the Double Chain Ladder method with a Bornhuetter-Ferguson adjustment. The Bornhuetter-Ferguson tecnhique is applied to stabilise the underwriting inflation parameters using incurred data

Usage

```
bdcl.estimation( Xtriangle , Ntriangle , Itriangle , adj = 1 ,
   Tables=TRUE , num.dec=4 , n.cal=NA , Fj.X=NA , Fj.N=NA , Fj.I=NA)
```

Arguments

`Xtriangle`	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
`Ntriangle`	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
`Itriangle`	The incurred triangle. It should be a matrix with incurred data located in the upper triangle. It is an incremental run-off triangle with the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
`adj`	Method to adjust the estimated delay parameters for the distributional model. It should be 1 (default value) or 2. See more in details below.
`Tables`	Logical. If TRUE (default) it is showed a table with the estimated parameters.
`num.dec`	Number of decimal places used to report numbers in the tables (if Tables=TRUE).
`n.cal`	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).
`Fj.X`	Optional vector with lentgth m-1 (m being the dimension of the triangles) with the development factors to calculate the chain ladder estimates from `Xtriangle`. See more details in `clm`.
`Fj.N`	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Ntriangle`.
`Fj.I`	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Itriangle`.

Details

Two model are estimated in the double chain ladder framework as with the `dcl.estimation` function. In this case the inflation parameter (`inflat`) is estimated from the incurred triangle (see BF adjustment in the description of the BDCL method in Martinez-Miranda, Nielsen and Verrall 2013). The predicted reserve using these estimates is different from the incurred reserve. If you want to reproduce exactly the incurred reserve (by splitting it into its RBNS and IBNR components) then use the function `idcl.estimation`.

Value

`pi.delay`	General delay parameters
`mu`	Mean severity factor
`inflat`	Underwriting severity inflation (BDCL inflation)

`inflat.DCL`	Underwriting severity inflation (DCL inflation)
`pj`	Delay probabilities (under a Multinomial assumption)
`mu.adj`	Adjusted mean factor corresponding to the `pj` parameters
`sigma2`	Variance severity factor
`phi`	Overdispersion parameter used to derive the estimate `sigma2`
`Ey`	Severity mean for each underwriting period
`Vy`	Severity variance for each underwriting period
`adj`	Type of adjusted used to derive the `pj` probabilities
`alpha.N`	Underwriting chain ladder parameter in the (OD)-Poisson model. Counts triangle (Ntriangle)
`beta.N`	Underwriting chain ladder parameter in the (OD)-Poisson model. Counts triangle (Ntriangle)
`Nhat`	The chain ladder preditions (counts triangle). It is a matrix having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).
`alpha.X`	Underwriting chain ladder parameter in the (OD)-Poisson model. Paid triangle (Xtriangle)
`beta.X`	Underwriting chain ladder parameter in the (OD)-Poisson model. Paid triangle (Xtriangle)
`Xhat`	The chain ladder preditions (paid triangle). It is a matrix having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).
`alpha.I`	Underwriting chain ladder parameter in the (OD)-Poisson model. Incurred triangle (Itriangle)
`beta.I`	Underwriting chain ladder parameter in the (OD)-Poisson model. Incurred triangle (Itriangle)

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North Americal Actuarial Journal*.

See Also

`get.incremental`, `Plot.dcl.par`, `dcl.predict`, `dcl.estimation`, `idcl.estimation`, `clm`

Examples

```
# Reproducing the data analysis in the paper by Martinez-Miranda, Nielsen and Verrall (2013)
data(NtriangleBDCL)
data(XtriangleBDCL)
data(ItriangleBDCL)

my.bdcl.par<-bdcl.estimation(XtriangleBDCL,NtriangleBDCL,ItriangleBDCL)
# Parameters shown in Table 1
Plot.dcl.par(my.bdcl.par,type.inflat='BDCL')
# BDCL Predictions by diagonals (future calendar years)
preds.bdcl.diag<-dcl.predict(my.bdcl.par,NtriangleBDCL,num.dec=0)
```

---

clm *Classical Chain Ladder Method*

Description

Provide the classical chain ladder output consisting of the development (forward) factors and the predictions in the full square. Besides it provides the estimated parameters under the (over-dispersion) Poisson model for the double chain ladder estimation.

Usage

```
clm( triangle , n.cal = NA , Fj = NA )
```

Arguments

`triangle`	The loss triangle. It should be a matrix with the observed counts (number of reported claims, number of payments etc.), aggregated payments or incurred data, located in the upper triangle. The lower triangle should consist of missing (NA) or zero values.
`n.cal`	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).
`Fj`	Optional. A vector with lentgth m-1 (m being the dimension of the triangle) with the development factors to calculate the chain ladder estimates. See more details below.

Details

By default `Fj=NA` and then classical chain ladder with the common calculation of the development factors (or using the most recent calendars -if 0<`n.cal`<m is provided), is performed. By specifying a valid vector with the development factors (it should has lenght equal to m-1), the user is allowed to use his own values in the algorithm. If valid values are specified for both `n.cal` and `Fj`, the first one (`n.cal`) will be ignored and the given development factors will be used in the calculations.

Value

triangle.hat	A matrix with dimension m having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).
alpha	Underwriting chain ladder parameter in the (OD)-Poisson model (see for example Verrall (1991) for a formal definition)
beta	Underwriting chain ladder parameter in the (OD)-Poisson model (Verrall 1991)
Fj	Development (forward) factors

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Verrall, R. (1991) Chain ladder and Maximum Likelihood. *Journal of the Institute of Actuaries* 118, 489-499.

See Also

get.incremental, Plot.clm.par, Plot.triangle

Examples

```
data(NtriangleDCL)
clm.N<-clm(NtriangleDCL)
# The alpha's
clm.N$alpha
# The beta's
clm.N$beta
# The development factors
clm.N$Fj
# Plotting the parameters and the dev. factors
Plot.clm.par(clm.N)
# The predictions
Nhat<-clm.N$triangle.hat
Plot.triangle(Nhat,Histogram=TRUE)

## Trying variations from classical chain ladder
# Try CLM only using the more recent 2 calendars in the development
# factors calculation
clm(NtriangleDCL,n.cal=2)

# Try CLM providing a vector with given development factors
my.Fj<-c(1.4,1.1,1.0,1.1,1.1,1.0,1.0,1.0,1.1)
clm(NtriangleDCL,Fj=my.Fj)
```

`dcl.boot` *Bootstrap distribution: the full cashflow*

Description

Provide the distribution of the IBNR, RBNS and total (RBNS+IBRN) reserves by calendar years and rows using bootstrapping.

Usage

```
dcl.boot( dcl.par , sigma2 , Ntriangle , boot.type = 2 , B = 999 ,
  Tail = TRUE , summ.by = "diag" , Tables = TRUE , num.dec = 2 , n.cal = NA)
```

Arguments

`dcl.par`	A list object with the estimated parameters: the value returned by the functions `dcl.estimation`, `bdcl.estimation` or `idcl.estimation`.
`sigma2`	Optional. The variance of the individual payments in the first underwriting period.
`Ntriangle`	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should be the same triangle used to get the value passed by the argument `dcl.par`.
`boot.type`	Choose between values 1, to provide only the variance process, or 2 (default), to take into account the uncertainty of the parameters.
`B`	The number of simulations in the bootstrap algorithm. The defaul value is 999.
`Tail`	Logical. If `TRUE` (default) the tail is provided.
`summ.by`	A character value such as `"diag"`, `"row"` or `"cell"`.
`Tables`	Logical. If `TRUE` (default) it is showed a table with the summary (mean, standard deviation, 1%, 5%, 50%, 95%, 99%) of the distribution of the outstanding liabilities in the future calendar periods (if `summ.by="diag"`) or by underwriting period (if `summ.by="row"`).
`num.dec`	Number of decimal places used to report numbers in the tables. Used only if `Tables=TRUE`
`n.cal`	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).

Details

If the calculated severity variance using the function `dcl.estimation` is not a valid estimate then it is recommended to provide directly this value through the argument `sigma2`. It can be calculated using the function `var` applied to a pilot sample of individual payments observed in the first underwriting period.

Value

`array.rbns.boot`
: An array with dimensions (m,2m-1,B) (m being the dimension of the input triangles in DCL). Each `array.rbns.boot[,,b]` is a matrix, with m rows and 2m-1 columns, having the bootstrapped outstanding RBNS numbers as the entries (for b=1,...,B).

`Mat.rbns`
: A matrix with B rows and 2m columns. Each `Mat.rbns[b,]` is a vector with elements being the outstanding liabilities for RBNS claims in the future calendar periods (sums by diagonals), and last value is the RBNS reserve (overall sum).

`array.ibnr.boot`
: An array with dimensions (m,2m-1,B) (m being the dimension of the input triangles in DCL). Each `array.ibnr.boot[,,b]` is a matrix, with m rows and 2m-1 columns, having the bootstrapped outstanding IBNR numbers as the entries (for b=1,...,B).

`Mat.ibnr`
: A matrix with B rows and 2m columns. Each `Mat.ibnr[b,]` is a vector with elements being the outstanding liabilities for IBNR claims in the future calendar periods (sums by diagonals), and last value is the RBNS reserve (overall sum).

`Mat.total`
: A matrix with B rows and 2m columns. Each `Mat.total[b,]` is a vector with elements being the outstanding liabilities for total(=RBNS+IBNR) claims in the future calendar periods (sums by diagonals), and last value is the RBNS reserve (overall sum).

`summ.rbns`
: A dataframe with the summary of the RBNS distribution. Only if `Tables=TRUE`.

`summ.ibnr`
: A dataframe with the summary of the IBNR distribution. Only if `Tables=TRUE`.

`summ.total`
: A dataframe with the summary of the total(=RBNS+IBNR) distribution. Only if `Tables=TRUE`.

Note

If `boot.type=2` the function will take some time to perform the calculations. It increases with the dimension of the triangles and the specified number of simulations `B`.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North Americal Actuarial Journal*.

See Also

`Plot.cashflow`, `dcl.boot.prior`

Examples

```
# Results described in the data application by Martinez-Miranda, Nielsen and Verrall (2012)
data(NtriangleDCL)
data(XtriangleDCL)

# Estimation of the DCL parameters
est<-dcl.estimation(XtriangleDCL,NtriangleDCL)

# Full cashflow considering the tail (only the variance process)
# Below only B=200 simulations to be faster in the example
boot1<-dcl.boot(dcl.par=est,Ntriangle=NtriangleDCL,boot.type=1,B=200)
Plot.cashflow(boot1)

# Full cashflow with tail and taking into account the parameters uncertainty
# and B=999 simulations. Do not run it unless you can wait about one minute
# boot2<-dcl.boot(dcl.par=est,Ntriangle=NtriangleDCL,boot.type=2)
# Plot.cashflow(boot2)
```

`dcl.boot.prior` *Bootstrap distribution (the full cashflow) adding prior knowledge*

Description

Provide the distribution of the IBNR, RBNS and total (RBNS+IBRN) reserves by calendar years and rows using bootstrapping.

Usage

```
dcl.boot.prior( Xtriangle , Ntriangle , sigma2 , mu , inflat.i , inflat.j , Qi ,
    Model = 2 , adj = 1 , boot.type = 2, B = 999 ,
    Tail = TRUE , summ.by = "diag", Tables = TRUE, num.dec = 2 , n.cal = NA ,
    Fj.X = NA , Fj.N = NA )
```

Arguments

`Xtriangle`	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
`Ntriangle`	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
`sigma2`	Optional. The variance of the individual payments in the first underwriting period.
`mu`	Optional. The mean of the individual payments in the first underwriting period.
`inflat.i`	Optional. A vector with dimension m (the dimension of the input triangles) specifying the severity inflation in the underwriting direction.

`inflat.j` Optional. A vector with dimension m specifying the severity inflation in the development direction. If not specified it will be assumed to be 1 and then the severity mean not depending on the development period.

`Qi` Optional. A vector with dimension m specifying the probability of zero-claims for each underwriting period. If not specified then it will be assumed no zero-payments.

`Model` Possible values are 0, 1 or 2 (default). See `dcl.estimation` for more in details.

`adj` Method to adjust the estimated delay parameters for the distributional model. It should be 1 (default value) or 2. See `dcl.estimation` for more details.

`boot.type` Choose between values 1, to provide only the variance process, or 2 (default), to take into account the uncertainty of the parameters.

`B` The number of simulations in the bootstrap algorithm. The defaul value is 999.

`Tail` Logical. If `TRUE` (default) the tail is provided.

`summ.by` A character value such as `"diag"`, `"row"` or `"cell"`.

`Tables` Logical. If `TRUE` (default) it is showed a table with the summary (mean, standard deviation, 1%, 5%, 50%, 95%, 99%) of the distribution of the outstanding liabilities in the future calendar periods (if `summ.by="diag"`) or by underwriting period (if `summ.by="row"`).

`num.dec` Number of decimal places used to report numbers in the tables. Used only if `Tables=TRUE`

`n.cal` Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).

`Fj.X` Optional vector with lentgth m-1 (m being the dimension of the triangles) with the development factors to calculate the chain ladder estimates from `Xtriangle`. See more details in `clm`.

`Fj.N` Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Ntriangle`.

Details

If proper values are provided for the arguments `sigma2`, `mu`, `inflat.i`, `inflat.j` and `Qi` then, they will be considered fixed as prior knowledge. Otherwise, if not specified, `inflat.j` will be assumed to be a vector of ones, `Qi` a vector of zeros, and the rest will be estimated using `dcl.estimation`.

Value

`array.rbns.boot`
An array with dimensions (m,2m-1,B) (m being the dimension of the input triangles in DCL). Each `array.rbns.boot[,,b]` is a matrix, with m rows and 2m-1 columns, having the bootstrapped outstanding RBNS numbers as the entries (for b=1,...,B).

`Mat.rbns` A matrix with B rows and 2m columns. Each `Mat.rbns[b,]` is a vector with elements being the outstanding liabilities for RBNS claims in the future calendar periods (sums by diagonals), and last value is the RBNS reserve (overall sum).

`array.ibnr.boot`
: An array with dimensions (m,2m-1,B) (m being the dimension of the input triangles in DCL). Each `array.ibnr.boot[,,b]` is a matrix, with m rows and 2m-1 columns, having the bootstrapped outstanding IBNR numbers as the entries (for b=1,...,B).

`Mat.ibnr`
: A matrix with B rows and 2m columns. Each `Mat.ibnr[b,]` is a vector with elements being the outstanding liabilities for IBNR claims in the future calendar periods (sums by diagonals), and last value is the RBNS reserve (overall sum).

`Mat.total`
: A matrix with B rows and 2m columns. Each `Mat.total[b,]` is a vector with elements being the outstanding liabilities for total(=RBNS+IBNR) claims in the future calendar periods (sums by diagonals), and last value is the RBNS reserve (overall sum).

`summ.rbns`
: A dataframe with the summary of the RBNS distribution. Only if `Tables=TRUE`.

`summ.ibnr`
: A dataframe with the summary of the IBNR distribution. Only if `Tables=TRUE`.

`summ.total`
: A dataframe with the summary of the total(=RBNS+IBNR) distribution. Only if `Tables=TRUE`.

Note

If `boot.type=2` the function will take some time to perform the calculations. It increases with the dimension of the triangles and the specified number of simulations `B`.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

See Also

`Plot.cashflow`, `dcl.boot`

Examples

```
## Data application by in Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013)
data(NtrianglePrior)
data(NpaidPrior)
data(XtrianglePrior)

## Extract information about zero-claims and severity dev. inflation
my.priors<-extract.prior(XtrianglePrior,NpaidPrior,NtrianglePrior,Plots=FALSE)
my.inflat.j<-my.priors$inflat.j
my.Qi<-my.priors$Qi

## Bootstrap cashflow incorporating prior knowledge about
##      severity inflation and zero claims
```

```
# Only variance process
# Below only B=200 simulations for a fast example
dist.priorC.I<-dcl.boot.prior(NtrianglePrior,XtrianglePrior,
  inflat.j=my.inflat.j,Qi=my.Qi,adj=2,Tail=FALSE,boot.type=1,B=200)
Plot.cashflow(dist.priorC.I)

## Try to compare with DCL with no prior knowledge:
# Only variance process
# dist.dcl.I<-dcl.boot.prior(NtrianglePrior,XtrianglePrior,adj=2,
#    Tail=FALSE,boot.type=1)
# Plot.cashflow(dist.dcl.I)
```

`dcl.estimation` *Parameter estimation - Double Chain Ladder model*

Description

Compute the estimated parameters in the model (delay parameters, severity underwriting inflation, severity mean and variance) using the Double Chain Ladder method.

Usage

```
dcl.estimation( Xtriangle , Ntriangle , adj = 1 , Tables = TRUE ,
   num.dec = 4 , n.cal = NA , Fj.X=NA , Fj.N=NA )
```

Arguments

`Xtriangle`	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
`Ntriangle`	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
`adj`	Method to adjust the estimated delay parameters for the distributional model. It should be 1 (default value) or 2. See more in details below.
`Tables`	Logical. If `TRUE` (default) it is showed a table with the estimated parameters.
`num.dec`	Number of decimal places used to report numbers in the tables (if `Tables=TRUE`).
`n.cal`	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).
`Fj.X`	Optional vector with lentgth m-1 (m being the dimension of the triangles) with the development factors to calculate the chain ladder estimates from `Xtriangle`. See more details in `clm`.
`Fj.N`	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Ntriangle`.

Details

Two models are estimated in the double chain ladder framework (Martinez-Miranda, Nielsen and Verrall 2012).

The basic DCL model only makes assumption on the first moments (see assumptions M1-M3 in Section 2 of the paper). From the two input triangles (`Ntriangle`,`Xtriangle`) the parameters involved in this model are estimated: `pi.delay` (delay parameters that could be negative values and/or sum up above 1, by solving the linear system (7) in Section 3), `mu` (mean of the individual payments in the first underwriting period, from expression (9)), `inflat` (the underwriting severity mean inflation, from expression (8)), `alpha.N` and `beta.N` (the chain ladder parameters in the (OD)Poisson model for `Ntriangle` from expressions (10)-(12)). Using the estimated parameters in this simpler model the predicted outstanding numbers (calculated from equations (14) and (15)) are exactly the classical chain ladder predictions (see Theorem 1 in pp. 67).

The second model is a distributional model (assumptions D1-D4 in Section 5) which allows to provide the full cash-flow. In this model the parameters are adjusted to match with the assumptions: `pj` are delay probabilities resulting from adjusting the general parameters `pi.delay` (defined in expressions (21)-(22)), `mu.adj` is the corresponding adjusted mean factor and `sigma2` is the variance factor (in expression (24)). The function `dcl.estimation` suggest two different adjustments of the general `pi.delay`, the user should choose the adjustment which does not modify substantially the IBNR/RBNS split in the basic model (M1-M3), see Martinez-Miranda, Nielsen, Verrall and Wlthrich (2013) for a discussion.

Value

`pi.delay`	General delay parameters
`mu`	Mean severity factor
`inflat`	Underwriting severity inflation
`pj`	Delay probabilities (under a Multinomial assumption)
`mu.adj`	Adjusted mean factor corresponding to the `pj` parameters
`sigma2`	Variance severity factor
`phi`	Overdispersion parameter used to derive the estimate `sigma2`
`Ey`	Severity mean for each underwriting period
`Vy`	Severity variance for each underwriting period
`adj`	Type of adjusted used to derive the `pj` probabilities
`alpha.N`	Underwriting chain ladder parameter in the (OD)-Poisson model. Counts triangle (Ntriangle)
`beta.N`	Underwriting chain ladder parameter in the (OD)-Poisson model. Counts triangle (Ntriangle)
`Nhat`	The chain ladder preditions (counts triangle). It is a matrix having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).
`alpha.X`	Underwriting chain ladder parameter in the (OD)-Poisson model. Paid triangle (Xtriangle)
`beta.X`	Underwriting chain ladder parameter in the (OD)-Poisson model. Paid triangle (Xtriangle)

`Xhat` The chain ladder preditions (paid triangle). It is a matrix having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and W|thrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*.

See Also

`Plot.dcl.par`, `dcl.predict`, `bdcl.estimation`, `idcl.estimation`, `clm`

Examples

```
data(NtriangleDCL)
data(XtriangleDCL)
# Estimation of the DCL parameters described in Martinez-Miranda, Nielsen and Verrall (2012)
est1<-dcl.estimation(XtriangleDCL,NtriangleDCL)
Plot.dcl.par(est1)
# Compare two possible adjustmets to get distributional parameters
# est1 with adj=1
pj.1<-est1$pj
pi.delay<-est1$pi.delay
est2<-dcl.estimation(XtriangleDCL,NtriangleDCL,adj=2,Tables=FALSE)
pj.2<-est2$pj
data.frame(pi.delay=pi.delay,pj.adj.1=pj.1,pj.adj.2=pj.2)
```

`dcl.predict` *Pointwise predictions (RBNS/IBNR split)*

Description

Pointwise predictions by calendar years and rows of the outstanding liabilities. The predictions are splitted between RBNS and IBNR claims.

Usage

```
dcl.predict( dcl.par , Ntriangle , Model = 2 , Tail = TRUE ,
  Tables = TRUE , summ.by="diag", num.dec = 2 )
```

Arguments

- `dcl.par` — A list object with the estimated parameters: the value returned by the functions `dcl.estimation`, `bdcl.estimation` or `idcl.estimation`.
- `Ntriangle` — Optional. The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as the `Xtriangle` (both in the same aggregation level (quarters, years,etc.)) used to derive `dcl.par`
- `Model` — Possible values are 0, 1 or 2 (default). See more details below.
- `Tail` — Logical. If `TRUE` (default) the tail is provided.
- `Tables` — Logical. If `TRUE` (default) it is shown a table with the predicted outstanding liabilities in the future calendar periods (`summ.by="diag"`) or by underwriting period (`summ.by="row"`).
- `summ.by` — A character value such as `"diag"`, `"row"` or `"cell"`.
- `num.dec` — Number of decimal places used to report numbers in the tables. Used only if `Tables=TRUE`

Details

If `Model=0` or `Model=1` then the predictions are calculated using the DCL model parameters in assumptions M1-M3 (general delay parameters, see Martinez-Miranda, Nielsen and Verrall 2012). If `Model=2` the adjusted delay probabilities (distributional model D1-D4) are considered. By choosing `Model=0` the predictions are calculated ignoring the observed counts in `Ntriangle` (also if the `Ntriangle` is not specified). It should be specified to reproduce get the IBNR/RBNS split of classical paid chain ladder.

Choose `summ.by="diag"` to calculate the predicted outstanding liabilities in the future calendar periods (diagonal sums), `summ.by="row"` for sums by underwriting periods (row sums); or `summ.by="cell"` to get only the the individual cell predictions.

Value

- `Xrbns` — A matrix with dimension m by 2m-1 (m being the dimension of the input triangles in DCL) having the outstanding RBNS numbers as the entries.
- `Drbns` — A vector with dimension 2m-1 with elements being the outstanding liabilities for RBNS claims in the future calendar periods (sums by diagonals). The last value is the RBNS reserve (overall sum).
- `Rrbns` — A vector with dimension m with elements being the outstanding liabilities for RBNS claims at each underwriting period (sums by rows). The last value is the RBNS reserve (overall sum).
- `Xibnr` — A matrix with dimension m by 2m-1 (m being the dimension of the input triangles in DCL) having the outstanding IBNR numbers as the entries.
- `Dibnr` — A vector with dimension 2m-1 with elements being the outstanding liabilities for IBNR claims in the future calendar periods (sums by diagonals). The last value is the IBNR reserve (overall sum).

Ribnr	A vector with dimension m with elements being the outstanding liabilities for IBNR claims at each underwriting period (sums by rows). The last value is the RBNS reserve (overall sum).
Xtotal	A matrix with dimension m by 2m-1 (m being the dimension of the input triangles in DCL) having the outstanding total (=RBNS+IBNR) numbers as the entries.
Dtotal	A vector with dimension 2m-1 with elements being the outstanding liabilities for all claims in the future calendar periods (sums by diagonals). The last value is the total (=RBNS+IBNR) reserve (overall sum).
Rtotal	A vector with dimension m with elements being the outstanding liabilities for all claims at each underwriting period (sums by rows). The last value is the total (=RBNS+IBNR) reserve (overall sum).

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and W|thrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*.

See Also

dcl.estimation, bdcl.estimation, idcl.estimation, dcl.predict.prior

Examples

```
## Data application by in Martinez-Miranda, Nielsen and Verrall (2012)

data(NtriangleDCL)
data(XtriangleDCL)

# Estimation of the DCL parameters described
est<-dcl.estimation(XtriangleDCL,NtriangleDCL)

# with general delay parameters and ignoring Ntriangle to reproduce exactly chain ladder
pred1<-dcl.predict(dcl.par=est,Model=1,Tail=FALSE)

# with Modeled parameters (distributional Model) and ignoring Ntriangle
pred2<-dcl.predict(dcl.par=est,Model=2,Tail=FALSE)

# with Modeled parameters (distributional Model) using observed Ntriangle
pred3<-dcl.predict(dcl.par=est,Ntriangle=NtriangleDCL,Model=2,Tail=FALSE)

# providing the Tail, with Modeled parameters (distributional Model)
pred4<-dcl.predict(dcl.par=est,Ntriangle=NtriangleDCL,Model=2,Tail=TRUE)
```

dcl.predict.prior — *Pointwise predictions (RBNS/IBNR split) adding prior knowledge*

Description

Pointwise predictions by calendar years and rows of the outstanding liabilities. The predictions are splitted between RBNS and IBNR claims.

Usage

```
dcl.predict.prior( Ntriangle , Xtriangle , inflat.i , inflat.j , Qi ,
  Model = 2, adj = 2, Tail = FALSE, Tables = TRUE,
  summ.by = "diag", num.dec = 2 )
```

Arguments

`Ntriangle`	Optional. The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as the `Xtriangle` (both in the same aggregation level (quarters, years,etc.)) used to derive `dcl.par`
`Xtriangle`	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
`inflat.i`	Optional. A vector with dimension m (the dimension of the input triangles) specifying the severity inflation in the underwriting direction. If not specified it will be estimated using `dcl.estimation`.
`inflat.j`	Optional. A vector with dimension m specifying the severity inflation in the development direction. If not specified it will be assumed to be 1 and then the severity mean not depending on the development period.
`Qi`	Optional. A vector with dimension m specifying the probability of zero-claims for each underwriting period. If not specified then it will be assumed no zero-payments.
`Model`	Possible values are 0, 1 or 2 (default). See `dcl.estimation` for more in details.
`adj`	Method to adjust the estimated delay parameters for the distributional model. It should be 1 (default value) or 2. See `dcl.estimation` for more in details.
`Tail`	Logical. If `TRUE` (default) the tail is provided.
`Tables`	Logical. If `TRUE` (default) it is showed a table with the predicted outstanding liabilities in the future calendar periods (`summ.by="diag"`) or by underwriting period (`summ.by="row"`).
`summ.by`	A character value such as `"diag"`, `"row"` or `"cell"`.
`num.dec`	Number of decimal places used to report numbers in the tables. Used only if `Tables=TRUE`

Details

The predictions are calculated under the first moment assumptions in the DCL model (see M1-M3) in Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012). In this case the severity mean is specified as

`inflat.i * (1-Qi) * inflat.j * mu`

where `inflat.i`, `Qi`, `inflat.j` and `mu` are prior information specified by the user. With this specification, the prediction formula consists of the expectation (conditional expectation -if `Ntriangle` is given and `Model=0`) of the future (RBNS/IBNR) aggregated payments. See formulas (8)-(9) in the paper.

If the prior information is not provided the function will return the DCL predictions as `dcl.predict`. The information about `Qi`, `inflat.j` can be extracted through DCL using `extract.prior`.

Value

`Xrbns` A matrix with dimension m by 2m-1 (m being the dimension of the input triangles in DCL) having the outstanding RBNS numbers as the entries.

`Drbns` A vector with dimension 2m-1 with elements being the outstanding liabilities for RBNS claims in the future calendar periods (sums by diagonals). The last value is the RBNS reserve (overall sum).

`Rrbns` A vector with dimension m with elements being the outstanding liabilities for RBNS claims at each underwriting period (sums by rows). The last value is the RBNS reserve (overall sum).

`Xibnr` A matrix with dimension m by 2m-1 (m being the dimension of the input triangles in DCL) having the outstanding IBNR numbers as the entries.

`Dibnr` A vector with dimension 2m-1 with elements being the outstanding liabilities for IBNR claims in the future calendar periods (sums by diagonals). The last value is the IBNR reserve (overall sum).

`Ribnr` A vector with dimension m with elements being the outstanding liabilities for IBNR claims at each underwriting period (sums by rows). The last value is the RBNS reserve (overall sum).

`Xtotal` A matrix with dimension m by 2m-1 (m being the dimension of the input triangles in DCL) having the outstanding total (=RBNS+IBNR) numbers as the entries.

`Dtotal` A vector with dimension 2m-1 with elements being the outstanding liabilities for all claims in the future calendar periods (sums by diagonals). The last value is the total (=RBNS+IBNR) reserve (overall sum).

`Rtotal` A vector with dimension m with elements being the outstanding liabilities for all claims at each underwriting period (sums by rows). The last value is the total (=RBNS+IBNR) reserve (overall sum).

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

See Also

dcl.estimation, bdcl.estimation, idcl.estimation, dcl.predict,extract.prior

Examples

```
## Data application by in Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013)
data(NtrianglePrior)
data(NpaidPrior)
data(XtrianglePrior)

Ntriangle<-NtrianglePrior
Xtriangle<-XtrianglePrior
Npaid<-NpaidPrior

## Extract information about zero-claims and severity dev. inflation
my.priors<-extract.prior(Xtriangle,Npaid,Ntriangle)
my.inflat.j<-my.priors$inflat.j
my.Qi<-my.priors$Qi

# Reproducing the poinwise predicions (tables 3,4,5) in the paper
# Note: in the paper we did not use Ntriangle in the predictions
# when modelling the predictions are slightly different

## Prior A: only using development year inflation
m<-nrow(Ntriangle)
preds.prior.A.gen<-dcl.predict.prior(Ntriangle,Xtriangle,
          inflat.j=my.inflat.j,Qi=rep(0,m),Model=0,adj=1,
          Tail=FALSE,Tables=TRUE,summ.by="diag",num.dec=2)

preds.prior.A.mod<-dcl.predict.prior(Ntriangle,Xtriangle,
          inflat.j=my.inflat.j,Qi=rep(0,m),Model=2,adj=2,
          Tail=FALSE,Tables=TRUE,summ.by="diag",num.dec=2)

## Prior B: only using zero claims inflation
preds.prior.B.gen<-dcl.predict.prior(Ntriangle,Xtriangle,
          inflat.j=rep(1,m),Qi=my.Qi,Model=0,adj=1,
          Tail=FALSE,Tables=TRUE,summ.by="diag",num.dec=2)

preds.prior.B.mod<-dcl.predict.prior(Ntriangle,Xtriangle,
          inflat.j=rep(1,m),Qi=my.Qi,Model=2,adj=2,
          Tail=FALSE,Tables=TRUE,summ.by="diag",num.dec=2)

## Prior C: only using development inflation and zero claims inflation
preds.prior.C.gen<-dcl.predict.prior(Ntriangle,Xtriangle,
```

```
          inflat.j=my.inflat.j,Qi=my.Qi,Model=0,adj=1,
          Tail=FALSE,Tables=TRUE,summ.by="diag",num.dec=2)

preds.prior.C.mod<-dcl.predict.prior(Ntriangle,Xtriangle,
          inflat.j=my.inflat.j,Qi=my.Qi,Model=2,adj=2,
          Tail=FALSE,Tables=TRUE,summ.by="diag",num.dec=2)
```

---

extract.prior *Extracting information about zero-claims and severity inflation*

Description

A way of extracting information about zero-claims and severity development inflation through the DCL method applied to two counts triangles: number of payments and number of reported claims.

Usage

```
extract.prior(Xtriangle, Npaid, Ntriangle, Plots = TRUE , n.cal = NA ,
  Fj.X = NA , Fj.N = NA , Fj.Npaid = NA )
```

Arguments

Xtriangle	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
Npaid	A run-off (incremental) triangle with the number of payments. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
Ntriangle	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
Plots	Logical. If TRUE (default) it is showed a two by one plot showing the extracted severity inflation in the development direction and the probability of zero-claims for each underwriting period.
n.cal	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).
Fj.X	Optional vector with lentgth m-1 (m being the dimension of the triangles) with the development factors to calculate the chain ladder estimates from `Xtriangle`. See more details in `clm`.
Fj.Npaid	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Npaid`.
Fj.N	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Ntriangle`.

Details

The function implements the strategy proposed in the paper by Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013) to extract information for additional triangles (see "Section 5: An example showing how other data can be used to provide prior information in practice"). The derived severity inflation `inflat.j` does not extend to the tail. If you want provide the tail, by using `dcl.predict.prior`, the vector should be extended to have dimension 2m-1, otherwise the tail will be not provided (as was done in the cited paper).

Value

`inflat.j` A vector with dimension m with the extracted severity inflation in the development direction.

`Qi` A vector with dimension m with the extracted probability of zero-claims for undewriting period.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

See Also

`dcl.predict.prior`, `dcl.estimation`

Examples

```
## Data application in Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013)
data(NtrianglePrior)
data(NpaidPrior)
data(XtrianglePrior)

extract.prior(XtrianglePrior,NpaidPrior,NtrianglePrior)
```

`get.cumulative` *Cumulative triangle*

Description

Switch from an incremental to a cumulative triangle

Usage

```
get.cumulative( triangle )
```

Arguments

`triangle`	An incremental run-off triangle

Value

The cumulative triangle

Note

The methods in this the DCL package works normally on incremental triangles

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

See Also

`get.incremental`

Examples

```
data(NtriangleDCL)
get.cumulative(NtriangleDCL)
```

`get.incremental` *Incremental triangle*

Description

Switch from an cumulative to an incremental triangle

Usage

```
get.incremental( triangle )
```

Arguments

`triangle`	A cumulative run-off triangle

Value

The incremental triangle

Note

The methods in this the DCL package works normally on incremental triangles

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

See Also

get.cumulative

Examples

```
data(NtriangleDCL)
Ntriangle.cum<-get.cumulative(NtriangleDCL)
get.incremental(Ntriangle.cum)
```

---

idcl.estimation *Parameter estimation - DCL model reproducing the incurred reserve.*

Description

Estimate the parameters in the Double Chain Ladder model model: delay parameters, severity mean and variance. The inflation parameter is corrected using the incurred data to provide the incurred cashflow.

Usage

```
idcl.estimation( Xtriangle , Ntriangle , Itriangle , adj = 1 ,
    Tables = TRUE , num.dec = 4 , n.cal = NA ,
    Fj.X = NA , Fj.N = NA , Fj.I = NA)
```

Arguments

Xtriangle	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
Ntriangle	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as Xtriangle (both in the same aggregation level (quarters, years,etc.))
Itriangle	The incurred triangle. It should be a matrix with incurred data located in the upper triangle. It is an incremental run-off triangle with the same dimension as Xtriangle (both in the same aggregation level (quarters, years,etc.))
adj	Method to adjust the estimated delay parameters for the distributional model. It should be 1 (default value) or 2. See more in details below.
Tables	Logical. If TRUE (default) it is showed a table with the estimated parameters.
num.dec	Number of decimal places used to report numbers in the tables (if Tables=TRUE).

n.cal	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default n.cal=NA and all the observed calendars are used (classical chain ladder).
Fj.X	Optional vector with lentgth m-1 (m being the dimension of the triangles) with the development factors to calculate the chain ladder estimates from Xtriangle. See more details in clm.
Fj.N	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from Ntriangle.
Fj.I	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from Itriangle.

Details

Two model are estimated in the double chain ladder framework as with the dcl.estimation function. In this case the DCL inflation parameter estimated by dcl.estimation from Ntriangle and Xtriangle is adjusted so that the derived predicted reserve is equal to the incurred reserve. Use this estimation method if you want the RBNS/IBNR split the incurred reserve and the incurred full cashflow.

Value

pi.delay	General delay parameters
mu	Mean severity factor
inflat	Underwriting severity inflation (BDCL inflation)
inflat.DCL	Underwriting severity inflation (DCL inflation)
pj	Delay probabilities (under a Multinomial assumption)
mu.adj	Adjusted mean factor corresponding to the pj parameters
sigma2	Variance severity factor
phi	Overdispersion parameter used to derive the estimate sigma2
Ey	Severity mean for each underwriting period
Vy	Severity variance for each underwriting period
adj	Type of adjusted used to derive the pj probabilities
alpha.N	Underwriting chain ladder parameter in the (OD)-Poisson model. Counts triangle (Ntriangle)
beta.N	Underwriting chain ladder parameter in the (OD)-Poisson model. Counts triangle (Ntriangle)
Nhat	The chain ladder preditions (counts triangle). It is a matrix having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).
alpha.X	Underwriting chain ladder parameter in the (OD)-Poisson model. Paid triangle (Xtriangle)
beta.X	Underwriting chain ladder parameter in the (OD)-Poisson model. Paid triangle (Xtriangle)

`Xhat` The chain ladder preditions (paid triangle). It is a matrix having the chain ladder predictions in the future (lower triangle) and the fitted values in the past (upper triangle).

`alpha.I` Underwriting chain ladder parameter in the (OD)-Poisson model. Incurred triangle (Itriangle)

`beta.I` Underwriting chain ladder parameter in the (OD)-Poisson model. Incurred triangle (Itriangle)

`CL.I.i` Outstanding incurred numbers (row sums of the lower predicted triangle) from classical chain ladder on the incurred triangle.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76. Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North Americal Actuarial Journal*, 17(2), 101-113.

See Also

`Plot.dcl.par`, `dcl.predict`, `dcl.estimation`, `bdcl.estimation`

Examples

```
data(NtriangleBDCL)
data(XtriangleBDCL)
data(ItriangleBDCL)

my.idcl.par<-idcl.estimation(XtriangleBDCL,NtriangleBDCL,ItriangleBDCL)
# Parameters
Plot.dcl.par(my.idcl.par,type.inflat='IDCL')
# IDCL Predictions by diagonals (future calendar years)
preds.idcl.diag<-dcl.predict(my.idcl.par,NtriangleBDCL,num.dec=0)

# Comparing with the BDCL method
my.bdcl.par<-bdcl.estimation(XtriangleBDCL,NtriangleBDCL,ItriangleBDCL)
# Parameters shown in Table 1
Plot.dcl.par(my.bdcl.par,type.inflat='BDCL')
# BDCL Predictions by diagonals (future calendar years)
preds.bdcl.diag<-dcl.predict(my.bdcl.par,NtriangleBDCL,num.dec=0)
```

ItriangleBDCL *Incurred data (BDCL example)*

Description

Real motor data from a major insurer. It is a yearly run-off (incremental) triangle consisting of the incurred data during 19 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen and Verrall (2013).

Usage

```
data(ItriangleBDCL)
```

Format

Matrix with dimension 19 by 19: 19 undewriting years and 19 development years.

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Fergusson. *North American Actuarial Journal*, 17(2), 101-113.

Examples

```
data(ItriangleBDCL)
data(XtriangleBDCL)
m<-nrow(XtriangleBDCL)

clm.I<-clm(ItriangleBDCL)
alpha.I<-clm.I$alpha
# The total paid for each accident year in the past
Ri.X<-rowSums(XtriangleBDCL,na.rm=TRUE)
# Incurred outstanding numbers
Ri.CL.incurred<-alpha.I-Ri.X
Total.CL.incurred<- sum(Ri.CL.incurred,na.rm=TRUE)

## Compare with CL on paid data
clm.X<-clm(XtriangleBDCL)
Xhat<-as.matrix(clm.X$triangle.hat)
Ri.CL.paid<-rowSums(Xhat)-rowSums(XtriangleBDCL,na.rm=TRUE)
Total.CL.paid<- sum(Ri.CL.paid,na.rm=TRUE)

# the predictions by rows
data.frame(underw.year=c(1:m,"Total"),CLM.paid=c(Ri.CL.paid,Total.CL.paid),
           CLM.incurred=round(c(Ri.CL.incurred,Total.CL.incurred),4))

# now the predictions by diagonals
inflat.factor<-Ri.CL.incurred/Ri.CL.paid
inflat.factor[Ri.CL.paid==0]<-1
```

```
# the lower triangle from incurred chain ladder is defined as:
Ihat<-Xhat
for (i in 1:m) Ihat[i,]<-Xhat[i,]*inflat.factor[i]
# now the sums by diagonals
Diag.CL.paid<-sapply(split(Xhat, row(Xhat)+col(Xhat)), sum, na.rm=TRUE)
Dclm.paid<-c(Diag.CL.paid[-(1:m)])
Total.CL.paid<- sum(Dclm.paid,na.rm=TRUE)
Dclm.paid<-c(Dclm.paid,Total.CL.paid)

Diag.CL.inc<-sapply(split(Ihat, row(Ihat)+col(Ihat)), sum, na.rm=TRUE)
Dclm.inc<-c(Diag.CL.inc[-(1:m)])
Total.CL.inc<- sum(Dclm.inc,na.rm=TRUE)
Dclm.inc<-c(Dclm.inc,Total.CL.inc)

# A table with the chain ladder predictions (paid and incurred data)
data.frame(Future.years=c(1:(m-1),'Tot.'),
    clm.paid=round(Dclm.paid),clm.incurred=round(Dclm.inc))
```

`NpaidPrior` *Number of non-zero payments (adding prior knowledge example)*

Description

It is a yearly run-off (incremental) triangle consisting of the number of non-zero payments during 14 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013).

Usage

```
data(NpaidPrior)
```

Format

Matrix with dimension 14 by 14: 14 undewriting years and 14 development years.

Source

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

Examples

```
data(NpaidPrior)
Plot.triangle(NpaidPrior)
```

NtriangleBDCL *Number of reported claims (BDCL example)*

Description

Real motor data from a major insurer. It is a yearly run-off (incremental) triangle consisting of the number of reported claims during 19 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen and Verrall (2013).

Usage

```
data(NtriangleBDCL)
```

Format

Matrix with dimension 19 by 19: 19 undewriting years and 19 development years.

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Fergusson. *North American Actuarial Journal*, 17(2), 101-113.

Examples

```
data(NtriangleBDCL)

Plot.triangle(NtriangleBDCL, Histogram=TRUE)
Plot.triangle(NtriangleBDCL)

# Classical chain ladder method
clm(NtriangleBDCL)
```

NtriangleDCL *Number of reported claims (DCL example)*

Description

Real motor data from a major insurer. It is a yearly run-off (incremental) triangle consisting of the number of reported claims during 10 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen and Verrall (2012).

Usage

```
data(NtriangleDCL)
```

Format

Matrix with dimension 10 by 10: 10 undewriting years and 10 development years.

Source

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Examples

```
data(NtriangleDCL)

Plot.triangle(NtriangleDCL, Histogram=TRUE)
Plot.triangle(NtriangleDCL)

# Classical chain ladder method
clm(NtriangleDCL)
```

`NtrianglePrior` *Number of reported claims (adding prior knowledge example)*

Description

It is a yearly run-off (incremental) triangle consisting of the number of reported claims during 14 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013).

Usage

```
data(NtrianglePrior)
```

Format

Matrix with dimension 14 by 14: 14 undewriting years and 14 development years.

Source

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

Examples

```
data(NtrianglePrior)

Plot.triangle(NtrianglePrior, Histogram=TRUE)
Plot.triangle(NtrianglePrior)

# Classical chain ladder method
clm(NtrianglePrior)
```

`Plot.cashflow` *Plotting the full cashflow (bootstrap distribution)*

Description

Provide histograms and boxplots of the RBNS, IBNR and total(=RBNS+IBNR) cashflows. The boxplots corresponds to the distribution of the outstanding liabities in the future calendar periods. The histograms show the distribution of the reserve (overall total).

Usage

```
Plot.cashflow( cashflow )
```

Arguments

`cashflow`	A list object returned by the function `dcl.boot` or `dcl.boot.prior`.

Details

The cashflow should be derived by specifying the parameter `summ.by="diag"` in the function `dcl.boot` or `dcl.boot.prior`.

Value

No returned value.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North Americal Actuarial Journal*, 17(2), 101-113.

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

See Also

`dcl.boot`, `dcl.boot.prior`, `dcl.estimation`

Examples

```
# Results described in the data application by Martinez-Miranda, Nielsen and Verrall (2012)
data(NtriangleDCL)
data(XtriangleDCL)

# Estimation of the DCL parameters
est<-dcl.estimation(XtriangleDCL,NtriangleDCL)
# Full cashflow considering the tail (only variance process)
# Below only B=200 simulations for a fast example
boot1<-dcl.boot(dcl.par=est,Ntriangle=NtriangleDCL,boot.type=1,B=200)
Plot.cashflow(boot1)
```

`Plot.clm.par` *Plotting the estimated chain ladder parameters*

Description

Show a plot with the estimates of the chain ladder parameters and the development factors

Usage

```
Plot.clm.par( clm.par )
```

Arguments

`clm.par`	A list object with the estimated chain ladder parameters: the value returned by the functions `clm`.

Value

No returned value

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

See Also

`clm`, `dcl.estimation`, `Plot.triangle`

Examples

```
data(NtriangleDCL)
my.clm.par<-clm(NtriangleDCL)

Plot.clm.par(my.clm.par)
```

Plot.dcl.par *Plotting the estimated parameters in the DCL model*

Description

Show a two by two plot with the estimated parameters in the Double Chain Ladder model

Usage

```
Plot.dcl.par( dcl.par , type.inflat = 'DCL' )
```

Arguments

dcl.par	A list object with the estimated parameters: the value returned by the functions dcl.estimation, bdcl.estimation or idcl.estimation.
type.inflat	Method used to estimate the inflation . Possible values are: 'DCL' (default) if it was used dcl.estimation, 'BDCL' if bdcl.estimation, and 'IDCL' if idcl.estimation

Value

No returned value

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North Americal Actuarial Journal*, 17(2), 101-113.

See Also

dcl.estimation, bdcl.estimation, idcl.estimation

Examples

```
data(NtriangleDCL)
data(XtriangleDCL)

# Estimation of the DCL parameters described in Martinez-Miranda, Nielsen and Verrall (2012)
my.dcl.par<-dcl.estimation(XtriangleDCL,NtriangleDCL)
Plot.dcl.par(my.dcl.par)
```

`Plot.triangle` *Plotting an incremental run-off triangle*

Description

Graphical representaion of an incremental run-off triangle.

Usage

```
Plot.triangle( triangle , Histogram = FALSE , tit='' )
```

Arguments

`triangle`	The loss incremental triangle. It should be a matrix with the observed counts (number of reported claims, number of payments etc.), payments or incurred data, located in the upper triangle. The lower triangle consisting in missing or zero values.
`Histogram`	Logical. If TRUE then a histogram representing the triangle is shown. If FALSE (default) then a classical representation of the triangle is shown, this is, a matplot of the row sums from both incremental and cummulative triangle.
`tit`	Character. Title to be added to the plot

Details

A histogram representation of the histogram is consistent with the run-off triangles of counts such as the number of reported claims, number of payments, etc. See Martinez-Miranda, Nielsen, Sperlich and Verrall (2013) for a further explanation.

Value

No returned value

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Martinez-Miranda M.D., Nielsen, J.P., Sperlich, S., Verrall, R. (2013). Continuous Chain Ladder: Reformulating and generalizing a classical insurance problem. Experts Systems with Applications, 40(14), 5588-5603.

See Also

`get.incremental`, `clm`

Examples

```
## Plotting a counts triangle (number of reported claims)
data(NtriangleDCL)
Plot.triangle(NtriangleDCL, Histogram=TRUE,tit=expression(paste('Counts: ',N[ij])))
# Classical CL predictions
clm.N<-clm(NtriangleDCL)
Nhat<-clm.N$triangle.hat
# Compare the original histogram with the CL projections
Plot.triangle(Nhat, Histogram=TRUE,tit='CL Predictions')

## Plotting a paid triangle (number of reported claims)
data(XtriangleDCL)
Plot.triangle(XtriangleDCL)
```

`validating.incurred` *Back-test: testing against the experience*

Description

A back-test to validate incurred reserve (IDCL) against paid reserve (DCL) or the paid with a Bornhuetter-Fergusson adjustment (BDCL). The validation strategy consists of: (1) Cut ncut=1,2,. diagonals from the observed paid triangle. (2) Apply the three methods (DCL, BDCL and IDCL), and (3) compare forecasts and actual values.

Usage

```
validating.incurred( ncut = 0 , Xtriangle , Ntriangle , Itriangle ,
  Model = 0 , Plot.box = TRUE , Tables = TRUE , num.dec = 4 ,
  n.cal = NA , Fj.X = NA , Fj.N = NA , Fj.I = NA)
```

Arguments

`ncut`	The number of last periods (diagonals) to cut from the paid triangle. The default value is 0 (see details below).

Xtriangle	The paid run-off triangle: incremental aggregated payments. It should be a matrix with incremental aggregated payments located in the upper triangle and the lower triangle consisting in missing or zero values.
Ntriangle	The counts data triangle: incremental number of reported claims. It should be a matrix with the observed counts located in the upper triangle and the lower triangle consisting in missing or zero values. It should has the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
Itriangle	The incurred triangle. It should be a matrix with incurred data located in the upper triangle. It is an incremental run-off triangle with the same dimension as `Xtriangle` (both in the same aggregation level (quarters, years,etc.))
Model	Possible values are 0, 1 or 2 (default). See more details below.
Plot.box	Logical. If `TRUE` (default) it is shown a boxplot of the errors in the cells predictions.
Tables	Logical. If `TRUE` (default) it is shown a table with the errors in the cells, diagonals and overall total predicion.
num.dec	Number of decimal places used to report numbers in the tables. Used only if `Tables=TRUE`.
n.cal	Integer specifying the number of most recent calendars which will be used to calculate the development factors. By default `n.cal=NA` and all the observed calendars are used (classical chain ladder).
Fj.X	Optional vector with lentgth m-1 (m being the dimension of the triangles) with the development factors to calculate the chain ladder estimates from `Xtriangle`. See more details in `clm`.
Fj.N	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Ntriangle`.
Fj.I	Optional vector with lentgth m-1 with the development factors to calculate the chain ladder estimates from `Itriangle`.

Details

If `ncut=0` the test is not computed but a plot showing the difference among the three methods is shown. It is recommended to start with this step to have some insight about the problem. Note that the first part in the IDCL inflation is usually very volatile since no many outstanding liabitities arise from the first underwriting periods.

The predicion errors provided through the value `pe.vector` are calculated as follow:

For individual cells: `pe.cells` = `sum(ce.dif^2)` / `sum(ce.obs^2)`

with `ce.dif` being the vector with the differences between the predicted cells and the actual cells (`ce.obs`).

For diagonals: `pe.diags` = `sum(ca.dif^2)` / `sum(ca.obs^2)`

with `ca.dif` being the vector with the differences between the predicted calendars and the actual calendars (`ca.obs`).

For the total reserve: `pe.tot` = `sum(tot.dif^2)` / `sum(tot.obs^2)`

with `tot.dif` the absolute difference between the predicted reserve and the actual reserve (tot.obs).

Value

- `pe.vector` — A vector (length 10) with elements being (in the following order): `ncut`, the averaged errors predicing cells by DCL, BDCL and IDCL (see `pe.cells` in Details above), the three averaged errors by predicing diagonals (`pe.diags`), and the three errors by predicting the overal total (`pe.tot`).
- `Xdif` — A matrix with the individual cells errors (see `ce.dif` in Details above) for each method (by columns)
- `Inflat.DCL` — The estimated underwriting DCL inflation using `dcl.estimation`. Only provided if `ncut=0`.
- `Inflat.BDCL` — The estimated underwriting BDCL inflation using `bdcl.estimation`. Only provided if `ncut=0`.
- `Inflat.IDCL` — The estimated underwriting IDCL inflation using `idcl.estimation`. Only provided if `ncut=0`.

Note

To validate classical chain ladder on paid data against classical chain ladder on incurred data it should be used `Model=0` (see `dcl.predict`) for more details.

Author(s)

M.D. Martinez-Miranda, J.P. Nielsen and R. Verrall

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Ferguson. *North Americal Actuarial Journal*, 17(2), 101-113.

Martinez-Miranda M.D., Nielsen, J.P., Sperlich, S., Verrall, R. (2013). Continuous Chain Ladder: Reformulating and generalizing a classical insurance problem. Experts Systems with Applications, 40(14), 5588-5603.

See Also

`dcl.estimation`, `bdcl.estimation`, `idcl.estimation`, ,`dcl.predict`

Examples

```
data(NtriangleBDCL)
data(XtriangleBDCL)
data(ItriangleBDCL)

Ntriangle<-NtriangleBDCL
Xtriangle<-XtriangleBDCL
Itriangle<-ItriangleBDCL
## First compare the three methods to be validated (three different inflations)
validating.incurred(ncut=0,Xtriangle,Ntriangle,Itriangle)

## Now perform a backtest cutting up to four calendars backward
```

```
test.res<-matrix(NA,4,10)
par(mfrow=c(2,2),cex.axis=0.9,cex.main=1)
par(mar=c(1.5,1.5,1.5,1.5),oma=c(1,0.5,0.5,0.2),mgp=c(3,0.5,0))
for (i in 1:4)
{
  res<-validating.incurred(ncut=i,Xtriangle,Ntriangle,Itriangle,Tables=FALSE)
  test.res[i,]<-as.numeric(res$pe.vector)
}
test.res<-as.data.frame(test.res)
names(test.res)<-c("num.cut","pe.point.DCL","pe.point.BDCL","pe.point.IDCL",
"pe.calendar.DCL","pe.calendar.BDCL","pe.calendar.IDCL",
"pe.total.DCL","pe.total.BDCL","pe.total.IDCL")
print(test.res)
```

`XtriangleBDCL` *Paid data (BDCL example)*

Description

Real motor data from a major insurer. It is a run-off (incremental) triangle consisting of the aggregated payments during 19 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen and Verrall (2013).

Usage

```
data(XtriangleBDCL)
```

Format

Matrix with dimension 19 by 19: 19 undewriting years and 19 development years.

References

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2013) Double Chain Ladder and Bornhuetter-Fergusson. *North American Actuarial Journal*, 17(2), 101-113.

Examples

```
data(XtriangleBDCL)

Plot.triangle(XtriangleBDCL)
# Classical chain ladder method
clm(XtriangleBDCL)
```

`XtriangleDCL` *Paid data (DCL example)*

Description

Real motor data from a major insurer. It is a yearly run-off (incremental) triangle consisting of the aggregated payments during 10 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen and Verrall (2012).

Usage

```
data(XtriangleDCL)
```

Format

Matrix with dimension 10 by 10: 10 undewriting years and 10 development years.

Source

Martinez-Miranda, M.D., Nielsen, J.P. and Verrall, R. (2012) Double Chain Ladder. *Astin Bulletin*, 42/1, 59-76.

Examples

```
data(XtriangleDCL)
Plot.triangle(XtriangleDCL)
# Classical chain ladder method
clm(XtriangleDCL)
```

`XtrianglePrior` *Paid data (adding prior knowledge example)*

Description

It is a run-off (incremental) triangle consisting of the aggregated payments during 19 years. These data were used in the empirical illustration provided by Martinez-Miranda, Nielsen, Verrall and Wuthrich (2013).

Usage

```
data(XtrianglePrior)
```

Format

Matrix with dimension 14 by 14: 14 undewriting years and 14 development years.

Source

Martinez-Miranda, M.D., Nielsen, J.P., Verrall, R. and Wuthrich, M.V. (2013) Double Chain Ladder, Claims Development Inflation and Zero Claims. *Scandinavian Actuarial Journal*. In press.

Examples

```
data(XtrianglePrior)

Plot.triangle(XtrianglePrior)
# Classical chain ladder method
clm(XtrianglePrior)
```

Index