Package: reinsureR (via r-universe)

August 24, 2024

Type Package

Title Reinsurance Treaties Application

Version 0.1.0

Author Arnaud Buzzi

Maintainer Arnaud Buzzi <arnaud.buzzi@sia-partners.com>

Description Application of reinsurance treaties to claims portfolios. The package creates a class Claims whose objective is to store claims and premiums, on which different treaties can be applied. A statistical analysis can then be applied to measure the impact of reinsurance, producing a table or graphical output. This package can be used for estimating the impact of reinsurance on several portfolios or for pricing treaties through statistical analysis. Documentation for the implemented methods can be found in ``Reinsurance: Actuarial and Statistical Aspects'' by Hansjöerg Albrecher, Jan Beirlant, Jozef L. Teugels (2017, ISBN: 978-0-470-77268-3) and ``REINSURANCE: A Basic Guide to Facultative and Treaty Reinsurance'' by Munich Re (2010) <https://www.munichre.com/site/mram/get/documents_E96160999/mram/assetpool.mr_america/PDFs/3_Publications/reinsurance_basic_guide.pdf>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 2.2), data.table, dplyr, viridis, viridisLite, ggplot2, methods

RoxygenNote 6.0.1

Collate 'utils.R' 'SL-class.R' 'XL-class.R' 'QS-class.R' 'Claims-class.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-20 19:07:33 UTC

Contents

`apply_treaty` *Claims: apply a treaty*

Description

`apply_treaty` applies a treaty on an object of type `Claims` (`Claims-class`).

Usage

```
apply_treaty(claims, treaty)
```

Arguments

`claims`	`Claims` (`Claims-class`) object
`treaty`	`Treaty` (`QS-class` or `XL-class` or `SL-class`) object

Value

The updated `Claims` object

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
treaty_1 <- xl(ded = 100000, lim = 20000, aad = 5000,
               aal = 200000, prm = 0.01, rns = 1)
claims <- apply_treaty(claims, treaty_1)
```

`claims` *Claims: User constructor function*

Description

`claims` defines an object of class `Claims` (`Claims-class`).

Usage

```
claims(claims, premiums)
```

Arguments

`claims`	Data.frame. Claims table. Contains at least 2 columns, and may contain 2 optional columns: • `year`: year associated with the considered claim; • `portfolio`: Optional. Portfolio associated with the considered claim; • `simulId`: Optional. Simulation id of the considered claim. Useful for stochastic modelling; • `amount`: amount of the considered claim.
`premiums`	Data.frame. Premiums table. Contains at least 2 columns, and may contain 1 optional column: • `year`: year associated with the considered premium; • `portfolio`: Optional. Portfolio associated with the considered premium; • `amount`: amount of the considered premium.

Details

If portfolio or simulId are not given in the input tables, the default value is set to 0 in the `Claims` object.

Consistency needs to be insured between claims and premiums. For every year where a claim is entered, a premium must be registered. The same goes for the portfolios.

Portfolios differenciation is used to apply reinsurance treaties to different part of claims.

Simulation Ids are used in the case of stochastic simulations, in order to apply reinsurance treaties over simulated claims. Simulations Id are not used for premiums, which are mapped through year and portfolio. Be careful to have a unique premium for each couple year/portfolio.

For a unique combination of year, portfolio and simulId can be associated multiple events that will be taken into account when applying Excess of Loss reinsurance.

Value

An object of class `Claims` (`Claims-class`), initialized with the values given in input. Its basic methods are:

- `show`
- `draw`(draw): plotting function;
- `summy`(summy): plotting function;
- `get_claims`(get_claims): extract claim table from Claims object;
- `get_premiums`(get_premiums): extract premium table from Claims object;
- `get_commissions`(get_commissions): extract commissions table from Claims object;
- `get_reinstatements`(get_reinstatements): extract reinstatements table from Claims object;
- `get_treaties`(get_treaties): extract applied treaties list from Claims object.

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)

p <- aggregate(amount ~ year, c, sum)

claims <- claims(c, p)
```

Claims-class *Claims*

Description

An S4 class to represent Claims and Premiums for reinsurance computations.

Usage

```
## S4 method for signature 'Claims'
show(object)
```

Arguments

`object`	The object to display

Methods (by generic)

- show: show method

Slots

`clm` Data.table. Claims table. Contains at least 4 columns:

- `year`: year associated with the considered claim;
- `portfolio`: portfolio associated with the considered claim;
- `simulId`: simulation id of the considered claim. Useful for stochastic modelling;
- `amount`: amount of the considered claim.

`prm` Data.table. Premiums table. Contains at least 3 columns:

- `year`: year associated with the considered premium;
- `portfolio`: portfolio associated with the considered premium;
- `amount`: amount of the considered premium.

`rns` Data.table. Reinstatement amount table, that only concerns excess of loss treaties. Contains at least 2 columns:

- `year`: year associated with the considered reinstatements;
- `simulId`: simulation id of the considered reinstatement amount. Useful for stochastic modelling.

`com` Data.table. Commissions amount table, that only concerns quota share treaties. Contains at least 2 columns:

- `year`: year associated with the considered commission amount;
- `portfolio`: portfolio associated with the considered commission.

`trt` Vector. List of treaties applied to the Claims object.

`draw` *Claims: Plot function*

Description

`draw` produces a graphical representation of an object `Claims` (`Claims-class`).

Usage

```
draw(x, value = "all", moment = "gain", output = "boxplot")

## S4 method for signature 'Claims'
draw(x, value = "all", moment = "gain",
  output = "boxplot")
```

Arguments

x	The Claims object to represent.
value	Character. The value to consider plot, among: • claims • premiums • reinstatements • commissions • all: default value. Compute the profit.
moment	Character. Moment for analysis: • before: before reinsurance; • after: after reinsurance; • gain: default value. Difference in values by application of reinsurance.
output	Character. Type of graph to produce: • boxplot: default value. Boxplot by year • histogram: histogram over all years

Details

For boxplots, a red dot represent the mean value for each year.

Value

a plot

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
treaty_1 <- xl(ded = 100000, lim = 20000, aad = 5000,
               aal = 200000, prm = 0.01, rns = 1)
claims <- apply_treaty(claims, treaty_1)
draw(claims)
```

---

get_claims *Get claims*

Description

get_claims gets the claims table of a Claims (Claims-class) object.

Usage

```
get_claims(object)

## S4 method for signature 'Claims'
get_claims(object)
```

Arguments

object	The object to display

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
get_claims(claims)
```

---

get_commissions *Get commissions*

Description

get_commissions gets the commissions induced by QS treaties applied to Claims (Claims-class) object.

Usage

```
get_commissions(object)

## S4 method for signature 'Claims'
get_commissions(object)
```

Arguments

object	The object to display

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
treaty_1 <- qs(0.8, com = 0.25)
claims <- apply_treaty(claims, treaty_1)
get_commissions(claims)
```

get_premiums *Get premiums*

Description

get_premiums gets the premiums table of a Claims (Claims-class) object.

Usage

```
get_premiums(object)

## S4 method for signature 'Claims'
get_premiums(object)
```

Arguments

object	The object to display

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
get_premiums(claims)
```

get_reinstatements *Get reinstatements*

Description

get_reinstatements gets the reinstatements induced by XL treaties applied to Claims (Claims-class) object.

Usage

```
get_reinstatements(object)

## S4 method for signature 'Claims'
get_reinstatements(object)
```

Arguments

object	The object to display

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
treaty_1 <- xl(ded = 100000, lim = 20000, aad = 5000,
               aal = 200000, prm = 0.01, rns = 1)
claims <- apply_treaty(claims, treaty_1)
get_reinstatements(claims)
```

---

get_treaties *Get treaties*

Description

`get_treaties` gets the characteristics of treaties applied to `Claims` (`Claims-class`) object.

Usage

```
get_treaties(object)

## S4 method for signature 'Claims'
get_treaties(object)
```

Arguments

`object`	The object to display

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
treaty_1 <- qs(0.8, com = 0.25)
claims <- apply_treaty(claims, treaty_1)
get_treaties(claims)
```

qs *Quota Share Treaty: User constructor function*

Description

qs defines an object of class QS (QS-class).

Usage

```
qs(csn_clm, csn_prm = "auto", com = 0, ptf = "all")
```

Arguments

csn_clm	Numeric. Cession rate on claims. Should be between 0 and 1.
csn_prm	Numeric. Cession rate on premiums. Should be between 0 and 1. Default value set to csn_clm.
com	Numeric. Commission rate, applied on the part of premiums given to the reinsurer. Should be between 0 and 1. Default value set to 0.
ptf	Vector. List of portfolios on which the treaty is to be applied on. Default value set to all.

Value

An object of class QS (QS-class), initialized with the values given in input. Its basic methods are:

- show

Examples

```
treaty_1 <- qs(0.8, com = 0.25)
treaty_1
```

QS-class *Quota Share Treaty*

Description

An S4 class to represent a Quota Share Treaty

Usage

```
## S4 method for signature 'QS'
show(object)
```

Arguments

object	the object to display

Methods (by generic)

- show: Quota Share Treaty: show method

Slots

csn_clm	Numeric. Cession rate on claims. Should be between 0 and 1.
csn_prm	Numeric. Cession rate on premiums. Should be between 0 and 1.
com	Numeric. Commission rate, applied on the part of premiums given to the reinsurer. Should be between 0 and 1.
ptf	Vector. List of portfolios on which the treaty is to be applied on.
trt	Character. Always equal to "QS". Identifier for the type of treaty.

---

sl *Stop Loss Treaty: User constructor function*

Description

sl defines an object of class SL (SL-class), which represents a Stop Loss treaty.

Usage

```
sl(ded = Inf, lim = Inf, prm = 0, ptf = "all")
```

Arguments

ded	Numeric. Deductible amount of the treaty. Should be superior to 0.
lim	Numeric. Limit amount for the treaty. Should be superior to 0. May be equal to Inf.
prm	Numeric. Premium rate, which represents the proportion of the premium given to the reinsurer as price for the treaty. Should be between 0 and 1.
ptf	Vector. List of portfolios on which the treaty is to be applied on. Default value set to all.

Value

An object of class SL (SL-class), initialized with the values given in input. Its basic methods are:

- show

Examples

```
treaty_1 <- sl(ded = 100000, lim = 20000, prm = 0.01, ptf = "all")
treaty_1
```

SL-class *Stop Loss Treaty*

Description

An S4 class to represent a Stop Loss Treaty

Usage

```
## S4 method for signature 'SL'
show(object)
```

Arguments

object	the object to display

Methods (by generic)

- show: Stop Loss Treaty: show method

Slots

ded Numeric. Deductible amount of the treaty. Should be superior to 0.

lim Numeric. Limit amount for the treaty. Should be superior to 0. May be equal to Inf.

prm Numeric. Premium rate, which represents the proportion of the premium given to the reinsurer as price for the treaty. Should be between 0 and 1.

ptf Vector. List of portfolios on which the treaty is to be applied on.

trt Character. Always equal to "SL". Identifier for the type of treaty

summy *Claims: Summary function*

Description

summy summarizes the data contained in the object Claims (Claims-class).

Usage

```
summy(object, op = "mean")

## S4 method for signature 'Claims'
summy(object, op = "mean")
```

Arguments

object	The Claims object to represent.
op	Character. The aggregation operation over the simulIds, among: • mean: default value. • sd • median • min • max

Value

The summarized data

Examples

```
c <- data.frame(year = unlist(sapply(2000:2017, function(x) rep(x, rpois(1,3)))))
c$amount <- pmax(rnorm(nrow(c), 200000, 100000), 0)
p <- aggregate(amount ~ year, c, sum)
claims <- claims(c, p)
treaty_1 <- xl(ded = 100000, lim = 20000, aad = 5000,
               aal = 200000, prm = 0.01, rns = 1)
claims <- apply_treaty(claims, treaty_1)
summy(claims)
```

---

xl *Excess of Loss Treaty: User constructor function*

Description

xl defines an object of class XL (XL-class), which represents an Excess of Loss treaty.

Usage

```
xl(ded = Inf, lim = Inf, aal = Inf, aad = Inf, prm = 0,
  rns = "none", ptf = "all")
```

Arguments

ded	Numeric. Deductible amount of the treaty. Should be superior to 0.
lim	Numeric. Limit amount for the treaty. Should be superior to 0. May be equal to Inf.
aal	Numeric. Annual Aggregate Deductible amount of the treaty. Should be superior to 0.
aad	Numeric. Annual Aggregate Limit amount for the treaty. Should be superior to 0. May be equal to Inf.

prm	Numeric. Premium rate, which represents the proportion of the premium given to the reinsurer as price for the treaty. Should be between 0 and 1.
rns	Numeric vector. Reinstatement prices. Vector of lentgh equals to the number of reinstatements with each value equals the price of the reinstatement.
ptf	Vector. List of portfolios on which the treaty is to be applied on. Default value set to all.

Details

Reinstatements are the number of time the limit can be reconstructed. The vector given for this parameter will be an indication of the price for each reinstatement. For example, a `rns` value of `c(0, 1)` will give one free reinstatement and one reinstatement paid 100% of the premium before totally consuming the limit.

Value

An object of class `XL` (`XL-class`), initialized with the values given in input. Its basic methods are:

- `show`

Examples

```
treaty_1 <- xl(ded = 200000, lim = 20000, aad = 0,
               aal = 10000, prm = 0.01, rns = c(0, 1))
treaty_1
```

XL-class *Excess of Loss Treaty*

Description

An S4 class to represent an Excess of Loss Treaty

Usage

```
## S4 method for signature 'XL'
show(object)
```

Arguments

object	the object to display

Methods (by generic)

- show: Excess of Loss Treaty: show method

Slots

`ded` Numeric. Deductible amount of the treaty. Should be superior to 0.

`lim` Numeric. Limit amount for the treaty. Should be superior to 0. May be equal to `Inf`.

`aal` Numeric. Annual Aggregate Deductible amount of the treaty. Should be superior to 0.

`aad` Numeric. Annual Aggregate Limit amount for the treaty. Should be superior to 0. May be equal to `Inf`.

`rns` Numeric vector. Reinstatement prices. Vector of lentgh equals to the number of reinstatements with each value equals the price of the reinstatement.

`prm` Numeric. Premium rate, which represents the proportion of the premium given to the reinsurer as price for the treaty. Should be between 0 and 1.

`ptf` Vector. List of portfolios on which the treaty is to be applied on.

`trt` Character. Always equal to "SL". Identifier for the type of treaty

Index