Description  A (not yet exhaustive) collection of common models of risk processes in actuarial science, represented as formal S4 classes. Each class (risk model) has a simulator of its path, and a plotting function. Further, a Monte-Carlo estimator of a ruin probability for a finite time is implemented, using a parallel computation. Currently, the package extends two classical risk models Cramer-Lundberg and Sparre Andersen models by including capital injections, that are positive jumps (see Breuer L. and Badescu A.L. (2014) <doi:10.1080/03461238.2011.636969>). The intent of the package is to provide a user-friendly interface for ruin processes' simulators, as well as a solid and extensible structure for future extensions.

License GPL-3

Entry description for 'ruin' package

Type  Package

Title  Simulation of Various Risk Processes

Version  0.1.1

Date  2018-07-20

Description  A (not yet exhaustive) collection of common models of risk processes in actuarial science, represented as formal S4 classes. Each class (risk model) has a simulator of its path, and a plotting function. Further, a Monte-Carlo estimator of a ruin probability for a finite time is implemented, using a parallel computation. Currently, the package extends two classical risk models Cramer-Lundberg and Sparre Andersen models by including capital injections, that are positive jumps (see Breuer L. and Badescu A.L. (2014) <doi:10.1080/03461238.2011.636969>). The intent of the package is to provide a user-friendly interface for ruin processes' simulators, as well as a solid and extensible structure for future extensions.

License GPL-3

Encoding UTF-8

LazyData true

URL http://github.com/irudnyts/ruin

BugReports http://github.com/irudnyts/ruin/issues

Depends  R (>= 3.5.0)

Imports  methods, parallel, ggplot2 (>= 2.2.1)

Collate  'AllClass.R' 'AllGeneric.R' 'methods-CramerLundberg.R'
  'methods-CramerLundbergCapitalInjections.R'
  'methods-SparreAndersen.R'
  'methods-SparreAndersenCapitalInjections.R' 'plot_path.R'
  'ruin_probability.R' 'zzz.R'

RoxygenNote  6.0.1

Suggests testthat, actuar (>= 2.3.0), knitr, rmarkdown

VignetteBuilder knitr
CramerLundberg

Description

CramerLundberg() constructs an object of CramerLundberg S4 class.

Usage

CramerLundberg(initial_capital = NULL, premium_rate = NULL,
                 claim_poisson_arrival_rate = NULL, claim_size_generator = NULL,
                 claim_size_parameters = NULL)
Arguments

initial_capital
a length one numeric non-negative vector specifying an initial capital. Default: 0.

premium_rate
a length one numeric non-negative vector specifying a premium rate. Default: 1.

claim_poisson_arrival_rate
a length one numeric positive vector specifying the rate of the Poisson process of claims’ arrivals. Default: 1.

claim_size_generator
a function indicating the random generator of claims’ sizes. Default: rexp.

claim_size_parameters
a named list containing parameters for the random generator of claims’ sizes. Default: list(rate = 1).

Details

The function constructs an object of a formal S4 class CramerLundberg, a representation of classical risk process defined as follows:

\[ X(t) = u + ct - \sum_{i=1}^{N(t)} Y_i, \]

where \( u \) is the initial capital (initial_capital), \( c \) is the premium rate (premium_rate), \( N(t) \) is the Poisson process with intensity \( \lambda \) (claim_poisson_arrival_rate), \( Y_i \) are iid claim sizes (claim_size_generator and claim_size_parameters).

Value

An object of CramerLundberg class.

References


See Also

CramerLundbergCapitalInjections, SparreAndersen, link{SparreAndersenCapitalInjections}.

Examples

model <- CramerLundberg(initial_capital = 10,
premium_rate = 1,
claim_poisson_arrival_rate = 1,
claim_size_generator = rexp,
claim_size_parameters = list(rate = 1))
A formal S4 class representation of classical Cramer-Lundberg model.

The model is defined as follows:

\[ X(t) = u + ct - \sum_{i=1}^{N(t)} Y_i, \]

where \( u \) is the initial capital (initial_capital), \( c \) is the premium rate (premium_rate), \( N(t) \) is the Poisson process with intensity \( \lambda \) (claim_poisson_arrival_rate), \( Y_i \) are iid claim sizes (claim_size_generator and claim_size_parameters).

Objects of class can be created only by using the constructor CramerLundberg.

Slots

- initial_capital: a length one numeric non-negative vector specifying an initial capital.
- premium_rate: a length one numeric non-negative vector specifying a premium rate.
- claim_poisson_arrival_rate: a length one numeric positive vector specifying the rate of the Poisson process of claims’ arrivals.
- claim_size_generator: a function indicating the random generator of claims’ sizes.
- claim_size_parameters: a named list containing parameters for the random generator of claims’ sizes.

References


See Also

CramerLundberg
CramerLundbergCapitalInjections

*Constructs an object of CramerLundbergCapitalInjections S4 class*

Description

CramerLundbergCapitalInjections() constructs an object of CramerLundbergCapitalInjections S4 class.

Usage

```r
CramerLundbergCapitalInjections(initial_capital = NULL, premium_rate = NULL,
                               claim_poisson_arrival_rate = NULL, claim_size_generator = NULL,
                               claim_size_parameters = NULL, capital_injection_poisson_rate = NULL,
                               capital_injection_size_generator = NULL,
                               capital_injection_size_parameters = NULL)
```

Arguments

- `initial_capital` a length one numeric non-negative vector specifying an initial capital. Default: 0.
- `premium_rate` a length one numeric non-negative vector specifying a premium rate. Default: 1.
- `claim_poisson_arrival_rate` a length one numeric positive vector specifying the rate of the Poisson process of claims’ arrivals. Default: 1.
- `claim_size_generator` a function indicating the random generator of claims’ sizes. Default: `rexp`.
- `claim_size_parameters` a named list containing parameters for the random generator of claims’ sizes. Default: `list(rate = 1)`.
- `capital_injection_poisson_rate` a length one numeric positive vector specifying the rate of the Poisson process of capital injections’ arrivals. Default: 1.
- `capital_injection_size_generator` a function indicating the random generator of capital injections’ sizes. Default: `rexp`.
- `capital_injection_size_parameters` a named list containing parameters for the random generator of capital injections’ sizes. Default: `list(rate = 1)`. 
Details

The function constructs an object of a formal S4 class CramerLundbergCapitalInjections, a representation of an extension of Cramer-Lundberg model that allows for positive jumps and defined as follows:

$$X(t) = u + ct + \sum_{k=1}^{N^+(t)} Y_k^+ - \sum_{i=1}^{N^-(t)} Y_i^-$$

where $u$ is the initial capital (initial_capital), $c$ is the premium rate (premium_rate), $N^+(t)$ is the Poisson process of positive jumps (capital injections) with intensity $\lambda^+(t)$ (capital_injection_poisson_rate), $Y_k^+$ are iid capital injections’ sizes (capital_injection_size_generator and capital_injection_size_parameters), $N^-(t)$ is the Poisson process of negative jumps (claims) with intensity $\lambda^-(t)$ (claim_poisson_arrival_rate), $Y_i^-$ are iid claim sizes (claim_size_generator and claim_size_parameters).

Value

An object of CramerLundbergCapitalInjections class.

References


See Also

CramerLundberg, SparreAndersen, link{SparreAndersenCapitalInjections}.

Examples

```r
model <- CramerLundbergCapitalInjections(
    initial_capital = 10,
    premium_rate = 1,
    claim_poisson_arrival_rate = 1,
    claim_size_generator = rexp,
    claim_size_parameters = list(rate = 1),
    capital_injection_poisson_rate = 1,
    capital_injection_size_generator = rexp,
    capital_injection_size_parameters = list(rate = 1)
)
```
CramerLundbergCapitalInjections-class

Details

The model is defined as follows:

\[
X(t) = u + ct + \sum_{k=1}^{N^+(t)} Y^+_k - \sum_{i=1}^{N^-(t)} Y^-_i
\]

where \( u \) is the initial capital (initial_capital), \( c \) is the premium rate (premium_rate), \( N^+(t) \) is the Poisson process of positive jumps (capital injections) with intensity \( \lambda^+ \) (capital_injection_poisson_rate), \( Y^+_k \) are iid capital injections' sizes (capital_injection_size_generator and capital_injection_size_parameters), \( N^-(t) \) is the Poisson process of negative jumps (claims) with intensity \( \lambda^- \) (claim_poisson_arrival_rate), \( Y^-_i \) are iid claim sizes (claim_size_generator and claim_size_parameters).

Objects of class can be created only by using the constructor `CramerLundbergCapitalInjections`.

Slots

initial_capital a length one numeric non-negative vector specifying an initial capital.

premium_rate a length one numeric non-negative vector specifying a premium rate.

claim_poisson_arrival_rate a length one numeric positive vector specifying the rate of the Poisson process of claims’ arrivals.

claim_size_generator a function indicating the random generator of claims’ sizes.

claim_size_parameters a named list containing parameters for the random generator of claims’ sizes.

capital_injection_poisson_rate a length one numeric positive vector specifying the rate of the Poisson process of capital injections’ arrivals.

capital_injection_size_generator a function indicating the random generator of capital injections’ sizes.

capital_injection_size_parameters a named list containing parameters for the random generator of capital injections’ sizes.

References


See Also

CramerLundbergCapitalInjections
PathCramerLundberg-class

A formal S4 class PathCramerLundberg

Description

A formal S4 class to contain a simulated path of CramerLundberg model.

Details

Objects of the class must not be created manually. Instead, objects of this class are returned by simulate_path.

Slots

model an object of CramerLundberg class.
path a numeric matrix of columns “time” and “X” defining the simulated path represented by pairs time-value.
claim_sizes a numeric vector of claims' sizes.
claim_arrival_times a numeric vector of claims’ interarrival times.
time_horizon a numeric vector of the maximum time horizon achieved.
is_ruined a logical vector indicating whether the process is ruined.
elapsed_time a numeric vector of the elapsed simulation time in seconds.
max_time_horizon a numeric vector of the maximum time horizon allowed.
max_simulation_time a numeric vector of the maximum simulation time in seconds.
seed a numeric vector indicating the seed used for simulation.

See Also

CramerLundberg (class definition), CramerLundberg (constructor).

PathCramerLundbergCapitalInjections-class

A formal S4 class PathCramerLundbergCapitalInjections

Description

A formal S4 class to contain a simulated path of CramerLundbergCapitalInjections model.

Details

Objects of the class must not be created manually. Instead, objects of this class are returned by simulate_path.
**PathSparreAndersen-class**

**Slots**

- `model` an object of `CramerLundbergCapitalInjections` class.
- `path` a numeric matrix of columns "time" and "X" defining the simulated path represented by pairs time-value.
- `claim_sizes` a numeric vector of claims' sizes.
- `claim_arrival_times` a numeric vector of claims' interarrival times.
- `capital_injection_sizes` a numeric vector of capital injections' sizes.
- `capital_injection_arrival_times` a numeric vector of capital injections' interarrival times.
- `time_horizon` a numeric vector of the maximum time horizon achieved.
- `is_ruined` a logical vector indicating whether the process is ruined.
- `elapsed_time` a numeric vector of the elapsed simulation time in seconds.
- `max_time_horizon` a numeric vector of the maximum time horizon allowed.
- `max_simulation_time` a numeric vector of the maximum simulation time in seconds.
- `seed` a numeric vector indicating the seed used for simulation.

**See Also**

- `CramerLundbergCapitalInjections` (class definition), `CramerLundbergCapitalInjections` (constructor).

---

**Description**

A formal S4 class to contain a simulated path of Sparre Andersen model.

**Details**

Objects of the class must not be created manually. Instead, objects of this class are returned by `simulate_path`.

**Slots**

- `model` an object of `CramerLundberg` class.
- `path` a numeric matrix of columns "time" and "X" defining the simulated path represented by pairs time-value.
- `claim_sizes` a numeric vector of claims' sizes.
- `claim_arrival_times` a numeric vector of claims' interarrival times.
- `time_horizon` a numeric vector of the maximum time horizon achieved.
- `is_ruined` a logical vector indicating whether the process is ruined.
PathSparreAndersenCapitalInjections-class

A formal S4 class PathSparreAndersenCapitalInjections

Description
A formal S4 class to contain a simulated path of SparreAndersenCapitalInjections model.

Details
Objects of the class must not be created manually. Instead, objects of this class are returned by simulate_path.

Slots
model an object of SparreAndersenCapitalInjections class.
path a numeric matrix of columns "time" and "X" defining the simulated path represented by pairs time-value.
claim_sizes a numeric vector of claims' sizes.
claim_arrival_times a numeric vector of claims' interarrival times.
capital_injection_sizes a numeric vector of capital injections' sizes.
capital_injection_arrival_times a numeric vector of capital injections' interarrival times.
time_horizon a numeric vector of the maximum time horizon achieved.
is_ruined a logical vector indicating whether the process is ruined.
elapsed_time a numeric vector of the elapsed simulation time in seconds.
max_time_horizon a numeric vector of the maximum time horizon allowed.
max_simulation_time a numeric vector of the maximum simulation time in seconds.
seed a numeric vector indicating the seed used for simulation.

See Also
SparreAndersenCapitalInjections (class definition), SparreAndersenCapitalInjections (constructor).
**plot_path**

Plot a path of the simulated ruin process

**Description**

plot_path() takes a simulated ruin process as the argument and plots its path.

**Usage**

plot_path(path_object)

**Arguments**

- **path_object** an S4 object of *Path class (e.g., PathCramerLundberg)*.

**Details**

Under the hood, the function uses ggplot2 package, therefore, all functionality from ggplot2 is available.

**Value**

A ggplot2 object.

**Examples**

```r
model <- CramerLundberg(initial_capital = 10,
                         premium_rate = 1,
                         claim_poisson_arrival_rate = 1,
                         claim_size_generator = rexp,
                         claim_size_parameters = list(rate = 1))

path <- simulate_path(model = model, max_time_horizon = 10)

plot_path(path)
```

**ruin_probability** Estimate a ruin probability for a finite time horizon

**Description**

ruin_probability simulates paths for a given risk model and returns a crude Monte-Carlo estimate of the ruin probability for the finite time horizon.
ruin_probability

Usage

ruin_probability(model, time_horizon, simulation_number = NULL,
                 ci_level = NULL, parallel = NULL, return_paths = NULL)

Arguments

model an S4 object indicating a risk model (e.g., CramerLundberg).
time_horizon a length one numeric finite vector specifying the time at which the ruin probability should be estimated.
simulation_number a length one numeric vector giving the number of simulations that should be performed. Default: 10000.

estimation

simulation_number

a length one numeric vector giving the number of simulations that should be performed. Default: 10000.

estimation

ci_level a length one numeric vector between 0 and 1 indicating the level of the confidence interval of the ruin probability. Default: 0.95.

parallel a length one logical vector indicating whether the parallel computing should be used. Default: TRUE.

return_paths a length one logical vector indicating whether a list of simulated paths should be returned. Default: FALSE.

Details

The function uses a parallel computing from the package parallel (if parallel is TRUE). The package sets up RNGkind to "L’Ecuyer-CMRG" for a safe seeding (see nextRNGStream) when it is loaded, so that user does not have to take care of seeds / RNGs. Further, the function allows computing the confidence interval, assuming the normal distribution of the ruin probability (thanks to CLT).

Value

A list of two elements: a numeric vector of lower bound of CI, estimate, and upper bound of CI of the ruin probability; and optionally the list of simulated paths.

Examples

model <- CramerLundberg(initial_capital = 0,
                        premium_rate = 1,
                        claim_poisson_arrival_rate = 1,
                        claim_size_generator = rexp,
                        claim_size_parameters = list(rate = 1))

ruin_probability(model = model,
                 time_horizon = 10,
                 simulation_number = 100,
                 return_paths = TRUE,
                 parallel = FALSE)
simulate_path

Generic for simulating a path of a given risk model

Description

simulate_path() simulates a path of a given risk model until one of the following conditions is met: (1) the process is ruined, (2) max_time_horizon is achieved, (3) the elapsed time of the simulation is greater than max_simulation_time.

Usage

simulate_path(model, max_time_horizon = NULL, max_simulation_time = NULL, seed = NULL)

Arguments

model an S4 object indicating a risk model (e.g., CramerLundberg).

max_time_horizon a length one numeric vector specifying the maximum time horizon, until with the process will be simulated. Default: Inf.

max_simulation_time a length one numeric vector indicating the maximum allowed time of simulation. The value should be specified in seconds. Default: Inf.

seed an optional arbitrary length numeric vector specifying the seed. If provided, the .Random.seed in .GlobalEnv is set to its value.

Value

An S4 corresponding to model class object. For instance, for CramerLundberg, the object of class PathCramerLundberg is returned.

Warning

Setting both max_time_horizon and max_simulation_time to Inf might be dangerous. In this case, the only stopping condition is a ruin of the process, which might not happen.

Examples

model <- CramerLundberg(initial_capital = 10,
premium_rate = 1,
claim_poisson_arrival_rate = 1,
claim_size_generator = rexp,
claim_size_parameters = list(rate = 1))

path <- simulate_path(model = model, max_time_horizon = 10)
simulate_path,CramerLundberg-method

Simulates a path of a Cramer-Lundberg model

Description

simulate_path() simulates a path of CramerLundberg model until one of the following conditions is met: (1) the process is ruined, (2) max_time_horizon is achieved, (3) the elapsed time of the simulation is greater than max_simulation_time.

Usage

```r
## S4 method for signature 'CramerLundberg'
simulate_path(model, max_time_horizon = NULL,
               max_simulation_time = NULL, seed = NULL)
```

Arguments

- `model` an S4 object of CramerLundberg class.
- `max_time_horizon` a length one numeric vector specifying the maximum time horizon, until with the process will be simulated. Default: Inf.
- `max_simulation_time` a length one numeric vector indicating the maximum allowed time of simulation. The value should be specified in seconds. Default: Inf.
- `seed` an optional arbitrary length numeric vector specifying the seed. If provided, the .Random.seed in .GlobalEnv is set to its value.

Value

PathCramerLundberg

Warning

Setting both max_time_horizon and max_simulation_time to Inf might be dangerous. In this case, the only stopping condition is a ruin of the process, which might not happen.

Examples

```r
model <- CramerLundberg(initial_capital = 10,
                         premium_rate = 1,
                         claim_poisson_arrival_rate = 1,
                         claim_size_generator = rexp,
                         claim_size_parameters = list(rate = 1))

path <- simulate_path(model = model, max_time_horizon = 10)
```
simulate_path,CramerLundbergCapitalInjections-method

Simulates a path of a Cramer-Lundberg model's extension with capital injections

Description

simulate_path() simulates a path of CramerLundbergCapitalInjections model until one of the following conditions is met: (1) the process is ruined, (2) max_time_horizon is achieved, (3) the elapsed time of the simulation is greater than max_simulation_time.

Usage

## S4 method for signature 'CramerLundbergCapitalInjections'
simulate_path(model,  
               max_time_horizon = NULL, max_simulation_time = NULL, seed = NULL)

Arguments

- **model**: an S4 object of CramerLundbergCapitalInjections class.
- **max_time_horizon**: a length one numeric vector specifying the maximum time horizon, until with the process will be simulated. Default: Inf.
- **max_simulation_time**: a length one numeric vector indicating the maximum allowed time of simulation. The value should be specified in seconds. Default: Inf.
- **seed**: an optional arbitrary length numeric vector specifying the seed. If provided, the .Random.seed in .GlobalEnv is set to its value.

Value

PathCramerLundbergCapitalInjections

Warning

Setting both max_time_horizon and max_simulation_time to Inf might be dangerous. In this case, the only stopping condition is a ruin of the process, which might not happen.

Examples

```r
model <- CramerLundbergCapitalInjections(
    initial_capital = 10,
    premium_rate = 1,
    claim_poisson_arrival_rate = 1,
    claim_size_generator = rexp,
    claim_size_parameters = list(rate = 1),
    capital_injection_poisson_rate = 1,
)```
capital_injection_size_generator = rexp,
capital_injection_size_parameters = list(rate = 2)
)

path <- simulate_path(model = model, max_time_horizon = 10)

simulate_path.SparreAndersen-method

Simulates a path of a Sparre Andersen model

Description

simulate_path() simulates a path of SparreAndersen model until one of the following conditions is met: (1) the process is ruined, (2) max_time_horizon is achieved, (3) the elapsed time of the simulation is greater than max_simulation_time.

Usage

## S4 method for signature 'SparreAndersen'
simulate_path(model, max_time_horizon = NULL,
             max_simulation_time = NULL, seed = NULL)

Arguments

model an S4 object of SparreAndersen class.
max_time_horizon

a length one numeric vector specifying the maximum time horizon, until with the process will be simulated. Default: Inf.
max_simulation_time

a length one numeric vector indicating the maximum allowed time of simulation. The value should be specified in seconds. Default: Inf.
seed

an optional arbitrary length numeric vector specifying the seed. If provided, the .Random.seed in .GlobalEnv is set to its value.

Value

PathSparreAndersen

Warning

Setting both max_time_horizon and max_simulation_time to Inf might be dangerous. In this case, the only stopping condition is a ruin of the process, which might not happen.
**Examples**

```r
code
model <- SparreAndersen(initial_capital = 10,
                        premium_rate = 1,
                        claim_interarrival_generator = rexp,
                        claim_interarrival_parameters = list(rate = 2),
                        claim_size_generator = rexp,
                        claim_size_parameters = list(rate = 1))

path <- simulate_path(model = model, max_time_horizon = 10)
```

**Description**

`simulate_path()` simulates a path of a Sparre Andersen model’s extension with capital injections until one of the following conditions is met: (1) the process is ruined, (2) `max_time_horizon` is achieved, (3) the elapsed time of the simulation is greater than `max_simulation_time`.

**Usage**

```r
## S4 method for signature 'SparreAndersenCapitalInjections'
simulate_path(model,
               max_time_horizon = NULL, max_simulation_time = NULL, seed = NULL)
```

**Arguments**

- `model`: an S4 object of `SparreAndersenCapitalInjections` class.
- `max_time_horizon`: a length one numeric vector specifying the maximum time horizon, until with the process will be simulated. Default: Inf.
- `max_simulation_time`: a length one numeric vector indicating the maximum allowed time of simulation. The value should be specified in seconds. Default: Inf.
- `seed`: an optional arbitrary length numeric vector specifying the seed. If provided, the `.Random.seed` in `.GlobalEnv` is set to its value.

**Value**

Path `SparreAndersenCapitalInjections`

**Warning**

Setting both `max_time_horizon` and `max_simulation_time` to Inf might be dangerous. In this case, the only stopping condition is a ruin of the process, which might not happen.
Examples

```r
code
model <- SparreAndersenCapitalInjections(
  initial_capital = 10,
  premium_rate = 1,
  claim_interarrival_generator = rexp,
  claim_interarrival_parameters = list(rate = 1),
  claim_size_generator = rexp,
  claim_size_parameters = list(rate = 1),
  capital_injection_interarrival_generator = rexp,
  capital_injection_interarrival_parameters = list(rate = 1),
  capital_injection_size_generator = rexp,
  capital_injection_size_parameters = list(rate = 2)
)

path <- simulate_path(model = model, max_time_horizon = 10)
```

SparreAndersen

constructs an object of SparreAndersen S4 class

Description

SparreAndersen() constructs an object of SparreAndersen S4 class.

Usage

```r
code
SparreAndersen(initial_capital = NULL, premium_rate = NULL,
  claim_interarrival_generator = NULL, claim_interarrival_parameters = NULL,
  claim_size_generator = NULL, claim_size_parameters = NULL)
```

Arguments

- `initial_capital`: a length one numeric non-negative vector specifying an initial capital. Default: 0.
- `premium_rate`: a length one numeric non-negative vector specifying a premium rate. Default: 1.
- `claim_interarrival_generator`: a function indicating the random generator of claims’ interarrival times. Default: rexp.
- `claim_interarrival_parameters`: a named list containing parameters for the random generator of claims’ interarrival times. Default: list(rate = 1).
- `claim_size_generator`: a function indicating the random generator of claims’ sizes. Default: rexp.
- `claim_size_parameters`: a named list containing parameters for the random generator of claims’ sizes. Default: list(rate = 1).
Details

The function constructs an object of a formal S4 class `SparreAndersen`, a representation of an extension of Cramer-Lundberg model that allows for non-exponential interarrival times and defined as follows:

\[ X(t) = u + ct - \sum_{i=1}^{N(t)} Y_i, \]

where \( u \) is the initial capital (`initial_capital`), \( c \) is the premium rate (`premium_rate`), \( N(t) \) is the renewal process defined by distribution of interarrival times (`claim_interarrival_generator` and `claim_interarrival_parameters`), \( Y_i \) are iid claim sizes (`claim_size_generator` and `claim_size_parameters`).

Value

An object of `SparreAndersen` class.

References


See Also

`CramerLundberg`, `CramerLundbergCapitalInjections`, `SparreAndersenCapitalInjections`.

Examples

```r
model <- SparreAndersen(
  initial_capital = 10,
  premium_rate = 1,
  claim_interarrival_generator = rexp,
  claim_interarrival_parameters = list(rate = 1),
  claim_size_generator = rexp,
  claim_size_parameters = list(rate = 1)
)
```

---

**Description**

A formal S4 class representation of classical Sparre Andersen model.
Details

The model is defined as follows:

\[ X(t) = u + ct - \sum_{i=1}^{N(t)} Y_i, \]

where \( u \) is the initial capital (initial_capital), \( c \) is the premium rate (premium_rate), \( N(t) \) is the renewal process defined by distribution of interarrival times (claim_interarrival_generator and claim_interarrival_parameters), \( Y_i \) are iid claim sizes (claim_size_generator and claim_size_parameters).

Objects of class can be created only by using the constructor SparreAndersen.

Slots

- **initial_capital**: a length one numeric non-negative vector specifying an initial capital.
- **premium_rate**: a length one numeric non-negative vector specifying a premium rate.
- **claim_interarrival_generator**: a function indicating the random generator of claims’ interarrival times.
- **claim_interarrival_parameters**: a named list containing parameters for the random generator of claims’ interarrival times.
- **claim_size_generator**: a function indicating the random generator of claims’ sizes.
- **claim_size_parameters**: a named list containing parameters for the random generator of claims’ sizes.

References


See Also

SparreAndersen

---

SparreAndersenCapitalInjections

*Constructs an object of SparreAndersenCapitalInjections S4 class*

Description

SparreAndersenCapitalInjections() constructs an object of SparreAndersenCapitalInjections S4 class.
SparreAndersenCapitalInjections

Usage

SparreAndersenCapitalInjections(initial_capital = NULL, premium_rate = NULL,
claim_interarrival_generator = NULL, claim_interarrival_parameters = NULL,
claim_size_generator = NULL, claim_size_parameters = NULL,
capital_injection_interarrival_generator = NULL,
capital_injection_interarrival_parameters = NULL,
capital_injection_size_generator = NULL,
capital_injection_size_parameters = NULL)

Arguments

initial_capital
a length one numeric non-negative vector specifying an initial capital. Default: 0.

premium_rate
a length one numeric non-negative vector specifying a premium rate. Default: 1.

claim_interarrival_generator
a function indicating the random generator of claims’ interarrival times. Default: rexp.

claim_interarrival_parameters
a named list containing parameters for the random generator of claims’ interarrival times. Default: list(rate = 1).

claim_size_generator
a function indicating the random generator of claims’ sizes. Default: rexp.

claim_size_parameters
a named list containing parameters for the random generator of claims’ sizes. Default: list(rate = 1).

capital_injection_interarrival_generator
a function indicating the random generator of capital injections’ interarrival times. Default: rexp.

capital_injection_interarrival_parameters
a named list containing parameters for the random generator of capital injections’ interarrival times. Default: list(rate = 1).

capital_injection_size_generator
a function indicating the random generator of capital injections’ sizes. Default: rexp.

capital_injection_size_parameters
a named list containing parameters for the random generator of capital injections’ sizes. Default: list(rate = 1).

Details

The function constructs an object of a formal S4 class SparreAndersenCapitalInjections, a representation of an extension of Sparre Andersen model that allows for positive jumps and defined as follows:

\[ X(t) = u + ct + \sum_{k=1}^{N^+(t)} Y_k^+ - \sum_{i=1}^{N^-(t)} Y_i^-

\]
where \( u \) is the initial capital (initial_capital), \( c \) is the premium rate (premium_rate), \( N^{(+)}(t) \) is the renewal process of positive jumps (capital injections) defined by distribution of interarrival times (capital_injection_interarrival_generator and capital_injection_interarrival_parameters), \( Y_k^{(+)} \) are iid capital injections' sizes (capital_injection_size_generator and capital_injection_size_parameters), \( N^{(-)}(t) \) is the renewal process of claims defined by distribution of interarrival times (claim_interarrival_generator and claim_interarrival_parameters), \( Y_i^{(-)} \) are iid claim sizes (claim_size_generator and claim_size_parameters).

Value

An object of SparreAndersenCapitalInjections class.

References


See Also

CramerLundberg, CramerLundbergCapitalInjections, link{SparreAndersen}.

Examples

```r
model <- SparreAndersenCapitalInjections(  
  initial_capital = 10,  
  premium_rate = 1,  
  claim_interarrival_generator = rexp,  
  claim_interarrival_parameters = list(rate = 1),  
  claim_size_generator = rexp,  
  claim_size_parameters = list(rate = 1),  
  capital_injection_interarrival_generator = rexp,  
  capital_injection_interarrival_parameters = list(rate = 1),  
  capital_injection_size_generator = rexp,  
  capital_injection_size_parameters = list(rate = 1)  
)
```

SparreAndersenCapitalInjections-class

A formal S4 class SparreAndersenCapitalInjections

Description

A formal S4 class representation of Sparre Andersen’s extension that includes capital injections.
Details

The model is defined as follows:

\[ X(t) = u + ct + \sum_{k=1}^{N^+(t)} Y_k^+ - \sum_{i=1}^{N^-(t)} Y_i^- \]

where \( u \) is the initial capital (initial_capital), \( c \) is the premium rate (premium_rate), \( N^+(t) \) is the renewal process of positive jumps (capital injections) defined by distribution of interarrival times (capital_injection_interarrival_generator and capital_injection_interarrival_parameters), \( Y_k^+ \) are iid capital injections’ sizes (capital_injection_size_generator and capital_injection_size_parameters), \( N^-(t) \) is the renewal process of claims defined by distribution of interarrival times (claim_interarrival_generator and claim_interarrival_parameters), \( Y_i^- \) are iid claim sizes (claim_size_generator and claim_size_parameters).

Objects of class can be created only by using the constructor \texttt{SparreAndersenCapitalInjections}.

Slots

- initial_capital: a length one numeric non-negative vector specifying an initial capital.
- premium_rate: a length one numeric non-negative vector specifying a premium rate.
- claim_interarrival_generator: a function indicating the random generator of claims’ interarrival times.
- claim_interarrival_parameters: a named list containing parameters for the random generator of claims’ interarrival times.
- claim_size_generator: a function indicating the random generator of claims’ sizes.
- claim_size_parameters: a named list containing parameters for the random generator of claims’ sizes.
- capital_injection_interarrival_generator: a function indicating the random generator of capital injections’ interarrival times.
- capital_injection_interarrival_parameters: a named list containing parameters for the random generator of capital injections’ interarrival times.
- capital_injection_size_generator: a function indicating the random generator of capital injections’ sizes.
- capital_injection_size_parameters: a named list containing parameters for the random generator of capital injections’ sizes.

References


See Also

\texttt{SparreAndersenCapitalInjections}
## Index

<table>
<thead>
<tr>
<th>Class/Function</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>CramerLundberg</td>
<td>2, 3, 4, 6, 8, 9, 12–14, 19, 22</td>
</tr>
<tr>
<td>CramerLundberg-class</td>
<td>4</td>
</tr>
<tr>
<td>CramerLundbergCapitalInjections</td>
<td>3, 5, 6–9, 15, 19, 22</td>
</tr>
<tr>
<td>CramerLundbergCapitalInjections-class</td>
<td>6</td>
</tr>
<tr>
<td>SparreAndersen</td>
<td>10, 17, 20, 22, 23</td>
</tr>
<tr>
<td>SparreAndersenCapitalInjections</td>
<td>10, 17, 20, 22, 23</td>
</tr>
<tr>
<td>SparreAndersenCapitalInjections-class</td>
<td>22</td>
</tr>
<tr>
<td>ggplot2</td>
<td>11</td>
</tr>
<tr>
<td>nextRNGStream</td>
<td>12</td>
</tr>
<tr>
<td>parallel</td>
<td>12</td>
</tr>
<tr>
<td>PathCramerLundberg</td>
<td>11, 13, 14</td>
</tr>
<tr>
<td>PathCramerLundberg-class</td>
<td>8</td>
</tr>
<tr>
<td>PathCramerLundbergCapitalInjections</td>
<td>15</td>
</tr>
<tr>
<td>PathCramerLundbergCapitalInjections-class</td>
<td>8</td>
</tr>
<tr>
<td>PathSparreAndersen</td>
<td>16</td>
</tr>
<tr>
<td>PathSparreAndersen-class</td>
<td>9</td>
</tr>
<tr>
<td>PathSparreAndersenCapitalInjections</td>
<td>17</td>
</tr>
<tr>
<td>PathSparreAndersenCapitalInjections-class</td>
<td>10</td>
</tr>
<tr>
<td>plot_path</td>
<td>11</td>
</tr>
<tr>
<td>RNGkind</td>
<td>12</td>
</tr>
<tr>
<td>ruin_probability</td>
<td>11</td>
</tr>
<tr>
<td>simulate_path</td>
<td>8–10, 13</td>
</tr>
<tr>
<td>simulate_path,CramerLundberg-method</td>
<td>14</td>
</tr>
<tr>
<td>simulate_path,CramerLundbergCapitalInjections-method</td>
<td>15</td>
</tr>
<tr>
<td>simulate_path,SparreAndersen-method</td>
<td>16</td>
</tr>
<tr>
<td>simulate_path,SparreAndersenCapitalInjections-method</td>
<td>17</td>
</tr>
<tr>
<td>SparreAndersen</td>
<td>3, 6, 9, 10, 16, 18, 19, 20</td>
</tr>
<tr>
<td>SparreAndersen-class</td>
<td>19</td>
</tr>
</tbody>
</table>