Package ‘CreditRisk’

July 21, 2025

Type Package

Title Evaluation of Credit Risk with Structural and Reduced Form Models

Version 0.1.7

Date 2024-04-18

Maintainer Alessandro Cimarelli <alessandro.cimarelli@icloud.com>

Description Evaluation of default probability of sovereign and corporate entities based on structural or intensity based models and calibration on market Credit Default Swap quotes. References: Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) <doi:10.1002/9781118818589>. Print ISBN: 9780470748466, Online ISBN: 9781118818589. © 2013 John Wiley & Sons Ltd.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports stats

RoxygenNote 7.3.1

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-19 10:53:09 UTC

Author Alessandro Cimarelli [aut, cre],
Nicolò Manca [aut]

Contents

`at1p` *Analytically - Tractable First Passage (AT1P) model*

Description

`at1p` calculates the survival probability $Q(\tau > t)$ and default intensity for each maturity according to the structural Analytically - Tractable First Passage model.

Usage

```
at1p(V0, H0, B, sigma, r, t)
```

Arguments

`V0`	firm value at time `t = 0` (it is a constant value).
`H0`	value of the safety level at time `t = 0`.
`B`	free positive parameter used for shaping the barrier `Ht`.
`sigma`	a vector of constant stepwise volatility σ_t.
`r`	a vector of constant stepwise risk-free rate.
`t`	a vector of debt maturity structure (it is a numeric vector).

Details

In this function the safety level `Ht` is calculated using the formula:

$$H(t) = \frac{H0}{V0} * E_0[V_t] * \exp^{-B\int_0^t \sigma_u du}$$

The backbone of the default barrier at t is a proportion, controlled by the parameter `H0`, of the expected value of the company assets at t. `H0` may depend on the level of the liabilities, on safety covenants, and in general on the characteristics of the capital structure of the company. Also, depending on the value of the parameter `B`, it is possible that this backbone is modified by accounting for the volatility of the company assets. For example, if `B > 0` corresponds to the interpretation that when volatility increases - which can be independed of credit quality - the barrier is slightly lowered to cut some more slack to the company before going bankrupt. When `B = 0` the barrier does not depend on the volatility and the "distance to default" is simply modelled through the barrier parameter `H0`.

Value

at1p returns an object of class data.frame containing the firm value, safety level $H(t)$ and the survival probability for each maturity. The last column is the default intensity calculated among each interval Δt.

References

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes.

Examples

```
mod <- at1p(V0 = 1, H0 = 0.7, B = 0.4, sigma = rep(0.1, 10), r = cdsdata$ED.Zero.Curve,
t = cdsdata$Maturity)
mod

plot(cdsdata$Maturity, mod$Ht, type = 'b', xlab = 'Maturity', ylab = 'Safety Level H(t)',
main = 'Safety level for different maturities', ylim = c(min(mod$Ht), 1.5),
col = 'red')
lines(cdsdata$Maturity, mod$Vt, xlab = 'Maturity', ylab = 'V(t)',
main = 'Value of the Firm \n at time t', type = 's')

plot(cdsdata$Maturity, mod$Survival, type = 'b',
main = 'Survival Probability for different Maturity \n (AT1P model)',
xlab = 'Maturity', ylab = 'Survival Probability')

matplot(cdsdata$Maturity, mod$Default.Intensity, type = 'l', xlab = 'Maturity',
ylab = 'Default Intensity')
```

BlackCox *Black and Cox's model*

Description

BlackCox calculates the survival probability $Q(\tau > t)$ and default intensity for each maturity according to the structural Black and Cox's model.

Usage

```
BlackCox(L, K = L, V0, sigma, r, gamma, t)
```

Arguments

L	debt face value at maturity t = T (it is a constant value).
K	positive parameter needed to calculate the safety level.
V0	firm value at time t = 0 (it is a constant value).
sigma	volatility (constant for all t).

r	risk-free rate (constant for all t).
gamma	interest rate used to discount the safety level Ht (it is a constant value).
t	a vector of debt maturity structure (it is a numeric vector).

Details

In Merton's model the default event can occurr only at debt maturity T while in Black and Cox's model the default event can occurr even before. In this model the safety level is given by the output Ht. Hitting this barrier is considered as an erlier default. Assuming a debt face value of L at the final maturity that coincides with the safety level in $t = T$, the safety level in $t \leq T$ is the K, with $K \leq L$, value discounted at back at time t using the interest rate gamma, obtaining:

$$H(t|t \leq T) = K * \exp^{-\gamma*(T-t)}$$

The output parameter Default.Intensity represents the default intensity of Δt. The firm's value Vt is calculated as in the Merton function.

Value

This function returns an object of class data.frame containing firm value, safety level $H(t)$ and the survival probability for each maturity. The last column is the default intensity calculated among each interval Δt.

References

David Lando (2004) Credit risk modeling.

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes.

Examples

```
mod <- BlackCox(L = 0.55, K = 0.40, V0 = 1, sigma = 0.3, r = 0.05, gamma = 0.04,
t = c(0.50, 1.00, 2.00, 5.00, 7.00, 10.00, 20.00, 30.00))
mod

plot(c(0.50, 1.00, 2.00, 5.00, 7.00, 10.00, 20.00, 30.00), mod$Ht, type = 'b',
     xlab = 'Maturity', ylab = 'Safety Level H(t)', main = 'Safety level for different
     maturities', ylim = c(min(mod$Ht), 1.5), col = 'red')
abline(h = 0.55, col = 'red')
lines(c(0.50, 1.00, 2.00, 5.00, 7.00, 10.00, 20.00, 30.00), mod$Vt, xlab = 'Maturity',
      ylab = 'V(t)', main = 'Value of the Firm \n at time t', type = 's')

plot(c(0.50, 1.00, 2.00, 5.00, 7.00, 10.00, 20.00, 30.00), mod$Survival, type = 'b',
     main = 'Survival Probability for different Maturity \n (Black & Cox model)',
     xlab = 'Maturity', ylab = 'Survival Probability')

matplot(c(0.50, 1.00, 2.00, 5.00, 7.00, 10.00, 20.00, 30.00), mod$Default.Intensity,
        type = 'l', xlab = 'Maturity', ylab = 'Default Intensity')
```

`calibrate.at1p` *AT1P model calibration to market CDS data*

Description

Compares CDS rates quoted on the market with theoric CDS rates calculeted by the function `cds` and looks for the parameters to be used into `at1p` for returning the default intensities corresponding to real market CDS rates performing the minimization of the objective function.

Usage

```
calibrate.at1p(V0, cdsrate, r, t, ...)
```

Arguments

`V0`	firm value at time `t = 0`.
`cdsrate`	CDS rates from market.
`r`	a vector of risk-free rate.
`t`	a vector of debt maturity structure.
`...`	additional parameters used in `cds` function.

Details

Inside `calibrate.at1p`, the function `objfn` takes the input a vector of parameters and returns the mean error occurred estimating CDS rates with `cds` function. The inputs used in `cds` are the default intensities calculated by the `at1p` function with the calibrated parameters. In particular the error is calculated as:

$$\frac{1}{n}\sum_{i=1}^{n}(c^{ds} - c^{ds}_{mkt})^2.$$

This quantity is a function of the default intensities and it is the objective function to be minimized in order to take optimal solutions for intensities.

Value

`calibrate.at1p` returns an object of class `data.frame` with calculated parameters of the `at1p` model and the error occurred in the minimization procedure.

References

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
calibrate.at1p(V0 = 1, cdsrate = cdsdata$Par.spread, r = cdsdata$ED.Zero.Curve,
t = cdsdata$Maturity)
```

calibrate.BlackCox *Black and Cox model calibration to market CDS data*

Description

Compares CDS rates quoted on the market with theoric CDS rates calculeted by the function cds and looks for the parameters to be used into BlackCox for returning the default intensities corresponding to real market CDS rates performing the minimization of the objective function.

Usage

```
calibrate.BlackCox(V0, cdsrate, r, t, ...)
```

Arguments

V0	firm value at time t = 0.
cdsrate	CDS rates from the market.
r	risk-free rate.
t	a vector of debt maturity structure.
...	additional parameters used in cds function.

Details

Inside calibrate.BlackCox, the function objfn takes the input a vector of parameters and returns the mean error occurred estimating CDS rates with cds function. The inputs used in cds are the default intensities calculated by the BlackCox function with the calibrated parameters. In particular the error is calculated as:

$$\frac{1}{n}\sum_{i=1}^{n}(c^{ds} - c^{ds}_{mkt})^2.$$

This quantity is a function of the default intensities and it is the objective function to be minimized in order to take optimal solutions for intensities.

Value

calibrate.BlackCox returns an object of class data.frame with calculated parameters of the BlackCox model and the error occurred in the minimization procedure.

References

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
calibrate.BlackCox(V0 = 1, cdsrate = cdsdata$Par.spread, r = 0.005, t = cdsdata$Maturity)
```

calibrate.cds — *Calibrate the default intensities to market CDS data*

Description

Compares CDS rates quoted on market with theoric CDS rates and looks for default intensities that correspond to real market CDS rates trough a minimization problem of an objective function.

Usage

```
calibrate.cds(r, t, Tj, cdsrate, ...)
```

Arguments

`r`	interest rates.
`t`	premiums timetable.
`Tj`	CDS maturities.
`cdsrate`	CDS rates from market.
`...`	additional parameters used in `cds` function.

Details

Inside `calibrate.cds`, the function `err.cds` takes the input a vector of intensities and return the mean error occurred estimating CDS rates with `cds`. In particular such error is calculated as:

$$\frac{1}{n}\sum_{i=1}^{n}(c^{ds} - c^{ds}_{mkt})^2.$$

This quantity is a function of default intensities and is the our objective function to be minimized in order to take optimal solutions for intensities.

Value

returns an object of class `list` with calculated intensities and the error occurred in the minimization procedure.

References

David Lando (2004) Credit risk modeling

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
calibrate.cds( r = cdsdata$ED.Zero.Curve, t = seq(.5, 30, by = 0.5),
            Tj = c(1, 2, 3, 4, 5, 7, 10, 20, 30), cdsrate = cdsdata$Par.spread, RR = 0.4)
```

calibrate.sbtv *SBTV model calibration to market CDS data*

Description

Compares CDS rates quoted on the market with theoric CDS rates calculeted by the function `cds` and looks for the parameters to be used into `sbtv` for returning the default intensities corresponding to real market CDS rates performing the minimization of the objective function.

Usage

```
calibrate.sbtv(V0, p, cdsrate, r, t, ...)
```

Arguments

`V0`	firm value at time `t = 0`.
`p`	vector of the probability of different scenario (sum of p must be 1).
`cdsrate`	CDS rates from market.
`r`	a vector of risk-free rate.
`t`	a vector of debt maturity structure.
`...`	additional parameters used in `cds` function.

Details

Inside `calibrate.sbtv`, the function `objfn` takes the input a vector of parameters and returns the mean error occurred estimating CDS rates with `cds` function. The inputs used in `cds` are the default intensities calculated by the `sbtv` function with the calibrated parameters. In particular the error is calculated as:

$$\frac{1}{n}\sum_{i=1}^{n}(c^{ds} - c^{ds}_{mkt})^2.$$

This quantity is a function of the default intensities and it is the objective function to be minimized in order to take optimal solutions for intensities.

Value

This function returns an object of class `list` with calculated parameters of `sbtv` model and the error occurred in the minimization procedure.

References

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
calibrate.sbtv(V0 = 1, p = c(0.95, 0.05), cdsrate = cdsdata$Par.spread,
r = cdsdata$ED.Zero.Curve, t = cdsdata$Maturity)
```

`cds` *Calculates Credit Default Swap rates*

Description

Calculates CDS rates starting form default intensities.

Usage

```
cds(t, int, r, R = 0.005, RR = 0.4, simplified = FALSE)
```

Arguments

`t`	premium timetable.
`int`	deterministic default intensities vector.
`r`	spot interest rates.
`R`	constant premium payments, value that the buyer pays in each t_i.
`RR`	recovery rate on the underline bond, default value is 40%.
`simplified`	logic argument. If FALSE calculates the CDS rates using the semplified version of calculations, if TRUE use the complete version.

Details

- Premium timetable is $t_i; i = 1, ..., T$. The vector starts from $t_1 \leq 1$, i.e. the first premium is payed at a year fraction in the possibility that the bond is not yet defaulted. Since premium are a postponed payment (unlike usual insurance contracts).
- Intensities timetable have domains $\gamma_i; i = t_1, ..., T$.
- spot interest rates of bond have domain $r_i; i = t_1, ..., T$. The function transforms spot rates in forward rates. If we specify that we want to calculate CDS rates with the simplified alghoritm, in each period, the amount of the constant premium payment is expressed by:

  $$\pi^{pb} = \sum_{i=1}^{T} p(0,i) S(0,i) \alpha_i$$

  and the amount of protection, assuming a recovery rate δ, is:

  $$\pi^{ps} = (1-\delta) \sum_{i=1}^{T} p(0,i) \hat{Q}(\tau = i) \alpha_i$$

If we want to calculate same quantities with the complete version, that evaluate premium in the continous, the value of the premium leg is calculated as:

$$\pi^{pb}(0,1) = -\int_{T_a}^{T_b} P(0,t) \cdot (t - T_{\beta(t)-1}) d_t Q(\tau \geq t) + \sum_{i=a+1}^{b} P(0,T_i) \cdot \alpha_i * Q(\tau \geq T_i)$$

and the protection leg as:

$$\pi^{ps}_{a,b}(1) := -\int_{t=T_a}^{T_b} P(0,t) d * Q(\tau \geq t)$$

In both versions the forward rates and intensities are supposed as costant stepwise functions with discontinuity in t_i

Value

`cds` returns an object of class `data.frame` with columns, for esch date t_i the value of survival probability, the premium and protection leg, CDS rate and CDS price.

References

David Lando (2004) Credit risk modeling.

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
cds(t = seq(0.5, 10, by = 0.5), int = seq(.01, 0.05, len = 20),
r = seq(0,0.02, len=20), R = 0.005, RR = 0.4, simplified = FALSE)
```

`cds2` *Calculate Credit Default Swap rates*

Description

Calculate CDS rates starting from default intensities

Usage

```
cds2(t, Tj, tr, r, tint, int, R = 0.005, ...)
```

Arguments

`t`	premium timetable.
`Tj`	CDS maturities.
`tr`	interest rates timetable.
`r`	spot interest rates.
`tint`	intensity timetable.
`int`	default intensities timetable.
`R`	constant premium payment.
`...`	further arguments on `cds`.

Details

The function `cds2` is based on `cds` but allows a more fine controll on maturities and on discretization of `r` and `int`. In particular input `(t, tr, tint)` can be of different length thanks to the function approx.

Value

An object of class `data.frame` that contains the quantities calculated by `cds` on Tj timetable.

References

David Lando (2004) Credit Risk Modeling.

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
cds2(t = c(1:20),Tj = c(1:20), tr = c(1:20), r = seq(0.01,0.06, len =20),
tint = c(1:20), int= seq(0.01,0.06, len =20))
```

`cdsdata` *CDS quotes from market*

Description

- `Maturity`: Maturities of cds contracts expressed in years;
- `Par.Spread`: CDS rates quotes, spread that nullify the present value of the two legs;
- `ED.Zero.Curve`: EURIBOR interest rates (risk-free)

Usage

```
data(cdsdata)
```

Format

An object of class "data.frame".

Source

Thomson Reuters, CDS quotes of Unicredit on 2017-01-23

---

cum_normal_density *Cumulative Normal Distribution Function*

Description

This function calculates the cumulative normal distribution function (CDF) for a given value x using the Hastings approximation method. This approximation is typically used in finance for the calculation of option pricing probabilities.

Usage

```
cum_normal_density(x)
```

Arguments

x	A numeric value or vector for which the cumulative normal distribution is to be calculated.

Details

The function uses a polynomial approximation as described by E.G. Haug in "The Complete Guide to Option Pricing Formulas" to estimate the CDF of a normal distribution. The coefficients used in the approximation are specifically chosen to minimize the error in the tail of the distribution, which is critical for financial applications like option pricing.

The polynomial approximation is applied to the normal density function:

$$N(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Then, the cumulative probability is adjusted based on the sign of x: - If x is non-negative, it returns \(1 - t\), where t is the polynomial approximation. - If x is negative, it returns \(t\).

The cumulative normal distribution function is important in statistics for hypothesis testing and in finance for the Black-Scholes option pricing formula.

Value

Returns the cumulative probability under the normal curve from \(-

$$\infty$$

\) to x.

References

Haug, E.G., The Complete Guide to Option Pricing Formulas. Hastings, C. Approximations for Digital Computers. Princeton Univ. Press, 1955.

Examples

```
cum_normal_density(1.96)
cum_normal_density(-1.96)
```

---

generalized_black_scholes
Generalized Black-Scholes Option Pricing Model

Description

This function calculates the price of a European call or put option using the generalized Black-Scholes formula, which extends the standard model to incorporate a continuous dividend yield.

Usage

```
generalized_black_scholes(TypeFlag = c("c", "p"), S, X, Time, r, b, sigma)
```

Arguments

`TypeFlag`	A character vector indicating the type of option to be priced, either "c" for call options or "p" for put options.
`S`	Current stock price (scalar).
`X`	Strike price of the option (scalar).
`Time`	Time to expiration of the option (in years).
`r`	Risk-free interest rate (annualized).
`b`	Cost of carry rate, b = r - q for a dividend yield q.
`sigma`	Volatility of the underlying asset (annualized).

Details

The generalized Black-Scholes formula considers both the risk-free rate and a cost of carry, making it suitable for a wider range of financial instruments, including commodities and currencies with continuous yields.

The pricing formula for call and put options is determined by:

$$C = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2)$$

$$P = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1)$$

where:

$$d_1 = \frac{\log(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

and $(N(\cdot))$ is the cumulative normal distribution function, estimated by the ‘cum_normal_density‘ function.

Value

Returns the price of the specified option (call or put).

References

Haug, E.G., The Complete Guide to Option Pricing Formulas.

Examples

```
# Calculate the price of a call option
generalized_black_scholes("c", S = 100, X = 100, Time = 1, r = 0.05, b = 0.05, sigma = 0.2)
# Calculate the price of a put option
generalized_black_scholes("p", S = 100, X = 100, Time = 1, r = 0.05, b = 0.05, sigma = 0.2)
```

`Merton` *Merton's model*

Description

`Merton` calculates the survival probability $Q(\tau > T)$ for each maturity according to the structural Merton's model.

Usage

```
Merton(L, V0, sigma, r, t)
```

Arguments

`L`	debt face value at maturity `t = T`; if the value of the firm V_T is below the debt face value to be paid in T the company default has occurred (it is a constant value).
`V0`	firm value at time `t = 0` (it is a constant value).
`sigma`	volatility (constant for all t).
`r`	risk-free rate (constant for all t).
`t`	a vector of debt maturity structure. The last value of this vector rapresents the debt maturity T.

Details

In Merton model the default event can occur only at debt maturity T and not before. In this model the debt face value `L` represents the constant safety level. In this model the firm value is the sum of the firm equity value `St` and ad the firm debt value `Dt`. The debt value at time $t < T$ is calculated by the formula:

$$D_t = L * \exp(-r(T-t)) - Put(t, T; V_t, L)$$

The equity value can be derived as a difference between the firm value and the debt:

$$S_t = V_t - D_t = V_t - L * \exp(-r(T-t)) + Put(t, T; V_t, L) = Call(t, T; V_t, L)$$

(by the put-call parity) so that in the Merton model the equity can be interpreted as a Call option on the value of the firm.

Value

`Merton` returns an object of class `data.frame` with:

- `Vt`: expected Firm value at time $t < T$ calculated by the simple formula $V_t = V_0 * \exp(rt)$.
- `St`: firm equity value at each $t < T$. This value can be seen as a call option on the firm value `V_t`.
- `Dt`: firm debt value at each $t < T$.
- `Survival`: survival probability for each maturity.

References

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes

Examples

```
mod <- Merton(L = 10, V0 = 20, sigma = 0.2, r = 0.005,
              t = c(0.50, 1.00, 2.00, 3.25, 5.00, 10.00, 15.00, 20.00))
mod

plot(c(0.50, 1.00, 2.00, 3.25, 5.00, 10.00, 15.00, 20.00), mod$Surv,
     main = 'Survival Probability for different Maturity \n (Merton model)',
     xlab = 'Maturity', ylab = 'Survival Probability', type = 'b')
```

`Merton.sim` *Firm value in Merton's model*

Description

With this function we simulate `n` trajectories of firm value based on Merton's model.

Usage

```
Merton.sim(V0, r, sigma, t, n, seed = as.numeric(Sys.time()))
```

Arguments

V0	firm value at time t = 0.
r	risk-free interest rate (constant for all t).
sigma	volatility (constant for all t).
t	a vector of debt maturity structure.
n	number of trajectories to be generated.
seed	starting seed, default seed is setted randomly.

Details

The trajectories are calculated according to the equation:

$$V_T = V_0 \exp \int_0^T dlnV_t$$

Where we express dln V_t using Ito's lemma to derive the differential of the logarithm of the firm value as:

$$dlnV_t = (\mu - \frac{\sigma^2}{2})dt + \sigma dW_t$$

Value

This function returns a matrix containing the simulated firm values.

References

Gergely Daròczi, Michael Puhle, Edina Berlinger, Péter Csòka, Dàniel Havran Màrton Michaletzky, Zsolt Tulasay, Kata Vàradi, Agnes Vidovics-Dancs (2013) Introduction to R for Quantitative Finance.

Examples

```
V <- Merton.sim(V0 = 20, r = 0.05, sigma = 0.2, t = seq(0, 30, by = 0.5), n = 5)
matplot(x = seq(0, 30, by = 0.5), y = V, type = 's', lty = 1, xlab = 'Time',
ylab = 'Firm value trajectories', main = "Trajectories of the firm values in the Merton's model")
```

---

sbtv — *Scenario Barrier Time-Varying Volatility AT1P model*

Description

sbtv calculates the survival probability $Q(\tau > t)$ and default intensity for each maturity according to the structural SBTV model.

Usage

```
sbtv(V0, H, p, B, sigma, r, t)
```

Arguments

`V0`	firm value at time `t = 0` (it is a constant value).
`H`	vector of differents safety level at time `t = 0`.
`p`	vector of the probability of different scenario (sum of p must be 1).
`B`	free positive parameter used for shaping the barrier `Ht`.
`sigma`	a vector of constant stepwise volatility σ_t.
`r`	a vector of constant stepwise risk-free rate.
`t`	a vector of debt maturity structure (it is a numeric vector).

Details

`sbtv` is an extension of the `at1p` model. In this model the parameter `H0` used in the `at1p` model is replaced by a random variable assuming different values in different scenarios, each scenario with a different probability. The survival probability is calculated as a weighted avarage of the survival probability using the formula:

$$SBTV.Surv = \sum_{i=1}^{N} p[i] * AT1P.Surv(H[i])$$

where `AT1P.Surv(H[i])` is the survival probability computed according to the AT1P model when $H_0 = H[i]$ and with weights equal to the probabilities of the different scenarios.

Value

`sbtv` returns an object of class `data.frame` containing the survival probability for each maturity. The last column is the default intensity calculated among each interval Δt.

References

Damiano Brigo, Massimo Morini, Andrea Pallavicini (2013) Counterparty Credit Risk, Collateral and Funding. With Pricing Cases for All Asset Classes.

Examples

```
mod <- sbtv(V0 = 1, H = c(0.4, 0.8), p = c(0.95, 0.05), B = 0, sigma = rep(0.20, 10),
            r = cdsdata$ED.Zero.Curve, t = cdsdata$Maturity)
mod

plot(cdsdata$Maturity, mod$Survival, type = 'b')
```

Index