

# Package: stochasticscenarios (via r-universe)

May 9, 2026

**Type** Package

**Title** A Package for Asset Projection

**Version** 1.4.0

**Date** 2025-09-13

**Author** T. Moudiki

**Maintainer** T. Moudiki <thierry.moudiki@gmail.com>

**Description** A Package for Asset Projection Using Stochastic Simulation

**License** GPL (>= 2)

**Depends** methods

**Encoding** UTF-8

**NeedsCompilation** no

**RoxygenNote** 7.3.2

**Repository** <https://techtonique.r-universe.dev>

**Date/Publication** 2025-09-13 03:33:00 UTC

**RemoteUrl** <https://github.com/Techtonique/stochasticscenarios>

**RemoteRef** HEAD

**RemoteSha** ac169a42919bdafadd5fd550312197ad86399384

## Contents

ESG-package . . . . .	2
Asset_PriceDistribution . . . . .	3
Bond_PriceDistribution . . . . .	4
CBond_PriceDistribution . . . . .	4
CDSPremium_PriceDistribution . . . . .	5
ConvBond_PriceDistribution . . . . .	5
customPathsGeneration . . . . .	6
EuroCall_Stock_PriceDistribution . . . . .	6
EuroCall_ZC_PriceDistribution . . . . .	7
EuroPut_Stock_PriceDistribution . . . . .	7

EuroPut_ZC_PriceDistribution . . . . .	7
getDefaultSpreadPaths . . . . .	8
getForwardRates . . . . .	8
getLiquiditySpreadPaths . . . . .	8
getParamsBaseScenarios . . . . .	8
getrealEstatePaths . . . . .	9
getRiskParamsScenarios . . . . .	9
getRiskParamsScenariosdefSpr . . . . .	9
getRiskParamsScenariosliqSpr . . . . .	10
getRiskParamsScenariosRE . . . . .	10
getRiskParamsScenariosrt . . . . .	10
getRiskParamsScenariosS . . . . .	11
getShortRatePaths . . . . .	11
getstockPaths . . . . .	11
getZCRates . . . . .	12
MartingaleTest . . . . .	12
ParamsScenarios . . . . .	13
rAllRisksFactors . . . . .	13
rAssetDistribution . . . . .	14
rDefaultSpread . . . . .	16
rLiquiditySpread . . . . .	16
rRealEstate . . . . .	17
rShortRate . . . . .	17
rStock . . . . .	18
Scenarios . . . . .	19
setForwardRates . . . . .	19
setParamsBaseScenarios . . . . .	20
setRiskParamsScenarios . . . . .	20
setRiskParamsScenariosdefSpr . . . . .	21
setRiskParamsScenariosliqSpr . . . . .	21
setRiskParamsScenariosRE . . . . .	22
setRiskParamsScenariosrt . . . . .	22
setRiskParamsScenariosS . . . . .	22
setZCRates . . . . .	23
ZC . . . . .	23
ZCBond_PriceDistribution . . . . .	24

<b>Index</b>	<b>25</b>
--------------	-----------

---

ESG-package

*ESG - Economic Scenario Generator*

---

## Description

Risk neutral Economic Scenario Generator.

**Details**

Package: ESG  
 Type: Package  
 Version: 1.2  
 Date: 2020-11-29  
 License: GNU  
 Depends: methods

The package is build around the "Scenarios" object.

Use the dedicated methods to set all the needed parameters then use the customPathsGeneration() method to proceed to the asset projection.

---

Asset\_PriceDistribution

*Asset\_PriceDistribution method*

---

**Description**

Get a distribution for any asset price. This method is a wrapper for asset specific pricers.

**Arguments**

type	The name of the asset to price. It must be 'Zero-Coupon', 'Bond', 'CBond', 'EuroCall_UL', 'EuroPut_UL', 'EuroCall_ZC' or 'EuroPut_ZC'.
t	Date of pricing (has to be an integer)
T	Date of maturity for the option
nCoupons	Number of coupons
couponsRate	Rate of coupons
omega	Recoverables in case of default
s	Date of maturity for the underlying
Strike	Strike for options

**Examples**

```

objScenario <- new("Scenarios")
# Basic scenario's parameters setting
objScenario <- setParamsBaseScenarios(objScenario, horizon = 10, nScenarios = 1000)
# Risk factors parameters setting
objScenario <- setRiskParamsScenariosrt(objScenario, vol = .1, k = 2)
objScenario <- setRiskParamsScenariosS(objScenario, vol = .1, k = 2,
volStock = .2, stock0 = 100, rho=.5)
objScenario <- setRiskParamsScenariosliqSpr(objScenario, eta=.05, liquiditySpread0=.01)
objScenario <- setRiskParamsScenariosdefSpr(objScenario, volDefault=.2,

```

```

defaultSpread0=.01, alpha=.1, beta=1)
# Forward and ZC rates setting
data(ZC)
objScenario <- setForwardRates(objScenario, ZC, horizon=10)
objScenario <- setZCRates(objScenario, ZC, horizon=10)
# Projection
objScenario <- customPathsGeneration(objScenario, type="shortRate")
objScenario <- customPathsGeneration(objScenario, type="stock")
objScenario <- customPathsGeneration(objScenario, type="defaultSpread")
objScenario <- customPathsGeneration(objScenario, type="liquiditySpread")
Asset_PriceDistribution(objScenario,type='ConvBond',t=0,T=10,nCoupons=1,couponsRate=0.03)

```

---

Bond\_PriceDistribution

*Bond\_PriceDistribution method*

---

### Description

Get a distribution for bond price.

### Arguments

t	Date of pricing (has to be an integer)
T	Date of maturity
nCoupons	Number of coupons
couponsRate	Rate of coupons

---

CBond\_PriceDistribution

*CBond\_PriceDistribution method*

---

### Description

Get a distribution for corporate bond price.

### Arguments

t	Date of pricing (has to be an integer)
T	Date of maturity
nCoupons	Number of coupons
couponsRate	Rate of coupons
omega	Recoverables in case of default

---

CDSPremium\_PriceDistribution  
*CDSPremium\_PriceDistribution*

---

### Description

Proceed to the projection using the parameters that were previously set into the Scenarios objet.

### Arguments

t	Date of pricing (has to be an integer)
T	Date of maturity for the option
omega	Recoverables in case of default

---

ConvBond\_PriceDistribution  
*ConvBond\_PriceDistribution method*

---

### Description

Proceed to the projection using the parameters that were previously set into the Scenarios objet.

### Arguments

type	The name of the asset for which a projection has to be proceeded. Can be 'short-Rate', 'stock', 'realEstate', 'liquiditySpread' or 'defaultSpread'. If NULL, all assets will be projected.
t	Date of pricing (has to be an integer)
T	Date of maturity for the option
nCoupons	Number of coupons
couponsRate	Rate of coupons

---

customPathsGeneration *customPathsGeneration method*

---

### Description

Proceed to the projection using the parameters that were previously set into the Scenarios objet.

### Arguments

type                    The name of the asset for which a projection has to be proceeded. Can be 'shortRate', 'stock', 'realEstate', 'liquiditySpread' or 'defaultSpread'. If NULL, all assets will be projected.

### Examples

```
objScenario <- new("Scenarios")
# Basic scenario's parameters setting
objScenario <- setParamsBaseScenarios(objScenario, horizon = 10, nScenarios = 1000)
# Risk factors parameters setting
objScenario <- setRiskParamsScenariosrt(objScenario, vol = .1, k = 2)
objScenario <- setRiskParamsScenariosS(objScenario, vol = .1, k = 2,
volStock = .2, stock0 = 100, rho=.5)
objScenario <- setRiskParamsScenariosliqSpr(objScenario, eta=.05, liquiditySpread=.01)
objScenario <- setRiskParamsScenariosdefSpr(objScenario, volDefault=.2,
defaultSpread=.01, alpha=.1, beta=1)
# Forward and ZC rates setting
data(ZC)
objScenario <- setForwardRates(objScenario, ZC, horizon=10)
objScenario <- setZCRates(objScenario, ZC, horizon=10)
# Projection
objScenario <- customPathsGeneration(objScenario, type="shortRate")
objScenario <- customPathsGeneration(objScenario, type="stock")
objScenario <- customPathsGeneration(objScenario, type="defaultSpread")
objScenario <- customPathsGeneration(objScenario, type="liquiditySpread")
```

---

EuroCall\_Stock\_PriceDistribution

*EuroCall\_Stock\_PriceDistribution method*

---

### Description

Get a distribution for EuroCall UL price.

### Arguments

t                        Date of pricing (has to be an integer)  
T                        Date of maturity  
Strike                    Strike of the option

---

EuroCall\_ZC\_PriceDistribution

*EuroCall\_ZC\_PriceDistribution method*

---

**Description**

Get a distribution for EuroCall ZC price.

**Arguments**

t	Date of pricing (has to be an integer)
T	Date of maturity
s	Date of maturity for the underlying
Strike	Strike of the option

---

EuroPut\_Stock\_PriceDistribution

*EuroPut\_Stock\_PriceDistribution method*

---

**Description**

Get a distribution for EuroPut UL price.

**Arguments**

t	Date of pricing (has to be an integer)
T	Date of maturity
Strike	Strike of the option

---

EuroPut\_ZC\_PriceDistribution

*EuroPut\_ZC\_PriceDistribution method*

---

**Description**

Get a distribution for EuroPut ZC price.

**Arguments**

t	Date of pricing (has to be an integer)
T	Date of maturity
s	Date of maturity for the underlying
Strike	Strike of the option

---

`getDefaultSpreadPaths` *getDefaultSpreadPaths method*

---

**Description**

Get default spread paths for a Scenarios object after projection.

---

`getForwardRates` *getForwardRates method*

---

**Description**

Get the forward rates for a Scenarios object.

**Examples**

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2, volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1, beta=1, eta=.05, rho=.5, stock0=100, realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
data(ZC)
scenarios1 <- setForwardRates(scenarios1, ZC, horizon=5)
getForwardRates(scenarios1)
```

---

`getLiquiditySpreadPaths`  
*getLiquiditySpreadPaths method*

---

**Description**

Get liquidity spread paths for a Scenarios object after projection.

---

`getParamsBaseScenarios`  
*getParamsBaseScenarios method*

---

**Description**

Get a list containing the horizon and number of scenarios for a Scenarios object.

**Examples**

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
getParamsBaseScenarios(scenarios1)
```

---

getrealEstatePaths     *getrealEstatePaths method*

---

### Description

Get real estate paths for a Scenarios object after projection.

---

getRiskParamsScenarios  
                                   *getRiskParamsScenarios method*

---

### Description

Get a list containing all risk paramaters for a Scenarios object.

### Examples

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
getRiskParamsScenarios(scenarios1)
```

---

getRiskParamsScenariosdefSpr  
                                   *getRiskParamsScenariosdefSpr method*

---

### Description

Get a list containing the risk paramaters related to default spread for a Scenarios object.

### Examples

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
getRiskParamsScenariosdefSpr(scenarios1)
```

---

```
getRiskParamsScenariosliqSpr
    getRiskParamsScenariosliqSpr method
```

---

**Description**

Get a list containing the risk paramaters related to the spread for a Scenarios object.

**Examples**

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
getRiskParamsScenariosliqSpr(scenarios1)
```

---

```
getRiskParamsScenariosRE
    getRiskParamsScenariosRE method
```

---

**Description**

Get a list containing the risk paramaters related to Real Estate for a Scenarios object.

**Examples**

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
getRiskParamsScenariosRE(scenarios1)
```

---

```
getRiskParamsScenariosrt
    getRiskParamsScenariosrt method
```

---

**Description**

Get a list containing the risk paramaters related to short rates for a Scenarios object.

**Examples**

```

scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
getRiskParamsScenariosrt(scenarios1)

```

---

getRiskParamsScenariosS

*getRiskParamsScenariosS method*


---

**Description**

Get a list containing the risk paramaters related to UL for a Scenarios object.

**Examples**

```

scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
getRiskParamsScenariosS(scenarios1)

```

---

getShortRatePaths

*getShortRatePaths method*


---

**Description**

Get the short rate paths for a Scenarios object after projection.

---

getstockPaths

*getstockPaths method*


---

**Description**

Get the UL paths for a Scenarios object after projection.

---

getZCRates	<i>getZCRates method</i>
------------	--------------------------

---

### Description

Get the ZC rates for a Scenarios object.

### Examples

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
data(ZC)
scenarios1 <- setZCRates(scenarios1, ZC, horizon=5)
getZCRates(scenarios1)
```

---

MartingaleTest	<i>MartingaleTest method</i>
----------------	------------------------------

---

### Description

Test the martingale for the Scenarios object.

### Examples

```
objScenario <- new("Scenarios")
# Basic scenario's parameters setting
objScenario <- setParamsBaseScenarios(objScenario, horizon = 10, nScenarios = 1000)
# Risk factors parameters setting
objScenario <- setRiskParamsScenariosrt(objScenario, vol = .1, k = 2)
objScenario <- setRiskParamsScenariosS(objScenario, vol = .1, k = 2,
volStock = .2, stock0 = 100, rho=.5)
objScenario <- setRiskParamsScenariosliqSpr(objScenario, eta=.05, liquiditySpread0=.01)
objScenario <- setRiskParamsScenariosdefSpr(objScenario, volDefault=.2,
defaultSpread0=.01, alpha=.1, beta=1)
# Forward and ZC rates setting
data(ZC)
objScenario <- setForwardRates(objScenario, ZC, horizon=10)
objScenario <- setZCRates(objScenario, ZC, horizon=10)
# Projection
objScenario <- customPathsGeneration(objScenario, type="shortRate")
objScenario <- customPathsGeneration(objScenario, type="stock")
objScenario <- customPathsGeneration(objScenario, type="defaultSpread")
objScenario <- customPathsGeneration(objScenario, type="liquiditySpread")
MartingaleTest(objScenario)
```

---

ParamsScenarios	<i>ParamsScenarios class</i>
-----------------	------------------------------

---

### Description

This class is a container for all the risk related parameters. It is used as a parameter for the Scenarios class.

### Details

**horizon** Horizon for the projection (in years)  
**nScenarios** Number of scenarios  
**vol** Volatility for rates in vasicek model  
**k** k for rates in vasicek model  
**volStock** Volatility for UL in Black & Scholes model  
**volRealEstate** Volatility for real estate in Black & Scholes model  
**stock0** Stock initial value  
**realEstate0** Real estate initial value  
**volDefault** Volatility for LMN model  
**alpha** alpha for LMN model  
**beta** beta for LMN model  
**eta** eta for LMN model  
**liquiditySpread0** Initial liquidity for LMN model  
**defaultSpread0** Initial default spread for LMN model  
**rho** Correlation between stock and short rates

---

rAllRisksFactors	<i>rAllRisksFactors</i>
------------------	-------------------------

---

### Description

Direct generation for all risk factors. Object creation is managed internally.

### Usage

```
rAllRisksFactors(horizon, nScenarios, ZC, vol, k,
    volStock, stock0, rho, volRealEstate, realEstate0, eta,
    liquiditySpread0, defaultSpread0, volDefault, alpha,
    beta)
```

**Arguments**

horizon	Horizon of projection
nScenarios	Number of scenarios
ZC	ZC rate input
vol	Volatility for short rates
k	k for rates in vasicek model
volStock	Volatility for stock
stock0	Initial value for stock
rho	Correlation between stock and short rates
volRealEstate	Volatility for real estate
realEstate0	Initial value for real estate
eta	eta Volatility for LMN model
liquiditySpread0	Initial value for liquidity spread
defaultSpread0	Initial value for default spread
volDefault	Volatility for default spread
alpha	alpha for LMN model
beta	beta Volatility for LMN model

**Examples**

```
data(ZC)
rAllRisksFactors(horizon=5, nScenarios=10, ZC, vol=.1, k=2, volStock=.2, stock0=100, rho=.5,
volRealEstate=.15, realEstate0=50, eta=.05, liquiditySpread0=.01, defaultSpread0=.01,
volDefault=.2, alpha=.1, beta=1)
```

---

`rAssetDistribution`     *rAssetDistribution*

---

**Description**

Direct generation for all assets values. Object creation is managed internally.

**Usage**

```
rAssetDistribution(type, t, T, vol, k, ZC,
nScenarios = NULL, volStock = NULL, stock0 = NULL,
rho = NULL, volRealEstate = NULL, realEstate0 = NULL,
eta = NULL, liquiditySpread0 = NULL,
defaultSpread0 = NULL, volDefault = NULL, alpha = NULL,
beta = NULL, nCoupons = NULL, couponsRate = NULL,
omega = NULL, s = NULL, Strike = NULL)
```

**Arguments**

type	Type of asset. Can be : Zero-Coupon, Bond, CBond, ConvBond, EuroCall_S, EuroPut_Stock, EuroCall_ZC, EuroPut_ZC or CDS.
t	Date of pricing (has to be an integer)
T	Date of maturity for the option
vol	Volatility for short rates
k	k for rates in vasicek model
ZC	ZC rate input
nScenarios	Number of scenarios
volStock	Volatility for stock
stock0	Initial value for stock
rho	Correlation between stock and short rates
volRealEstate	Volatility for real estate
realEstate0	Initial value for real estate
eta	eta Volatility for LMN model
liquiditySpread0	Initial value for liquidity spread
defaultSpread0	Initial value for default spread
volDefault	Volatility for default spread
alpha	alpha for LMN model
beta	beta Volatility for LMN model
nCoupons	Number of coupons
couponsRate	Rate of coupons
omega	Recoverables in case of default
s	Date of maturity for the underlying
Strike	Strike for options

**Examples**

```

data(ZC)
rAssetDistribution(type="Zero-Coupon",t=2,T=3,vol=.1, k=2, ZC=ZC, nScenarios=100)
rAssetDistribution(type="Bond",t=3,T=35,nCoupons=20, couponsRate=0.3,vol=.1, k=2,
ZC=ZC, nScenarios=10)
rAssetDistribution(type="CBond",t=5,T=35,nCoupons=5, couponsRate=0.3, omega=5,vol=.1, k=2, ZC=ZC,
nScenarios=10,eta=.05, liquiditySpread0=.01, defaultSpread0=.01, volDefault=.2, alpha=.1, beta=1)
rAssetDistribution(type="EuroPut_Stock",5,25,Strike=98.5,vol=.1,k=2,ZC=ZC,volStock=.2,
stock0=100, rho=.5,nScenarios=10)
rAssetDistribution(type="EuroCall_ZC",4,4.5,s=5, Strike=.985,vol=.1, k=2, ZC=ZC,nScenarios=10)
rAssetDistribution(type="EuroPut_ZC",4,4.5,s=5, Strike=.9385,vol=.1, k=2, ZC=ZC,nScenarios=10)

```

---

<code>rDefaultSpread</code>	<i>rDefaultSpread</i>
-----------------------------	-----------------------

---

**Description**

Direct default spread generation. Object creation is managed internally.

**Usage**

```
rDefaultSpread(horizon, nScenarios, defaultSpread0,
               volDefault, alpha, beta)
```

**Arguments**

<code>horizon</code>	Horizon of projection
<code>nScenarios</code>	Number of scenarios
<code>defaultSpread0</code>	Initial value for default spread
<code>volDefault</code>	Volatility
<code>alpha</code>	alpha for LMN model
<code>beta</code>	beta Volatility for LMN model

**Examples**

```
rDefaultSpread(horizon=5, nScenarios=8, defaultSpread0=.01, volDefault=.2, alpha=.1, beta=1)
```

---

<code>rLiquiditySpread</code>	<i>rLiquiditySpread</i>
-------------------------------	-------------------------

---

**Description**

Direct liquidity spread generation. Object creation is managed internally.

**Usage**

```
rLiquiditySpread(horizon, nScenarios, eta,
                 liquiditySpread0)
```

**Arguments**

<code>horizon</code>	Horizon of projection
<code>nScenarios</code>	Number of scenarios
<code>eta</code>	eta Volatility for LMN model
<code>liquiditySpread0</code>	Initial value for liquidity spread

**Examples**

```
rLiquiditySpread(horizon=5, nScenarios=15, eta=.05, liquiditySpread0=.01)
```

---

rRealEstate

*rRealEstate*


---

**Description**

Direct real estate generation. Object creation is managed internally.

**Usage**

```
rRealEstate(horizon, nScenarios, ZC, vol, k,
            volRealEstate, realEstate0)
```

**Arguments**

horizon	Horizon of projection
nScenarios	Number of scenarios
ZC	ZC rate input
vol	Volatility for short rates
k	k for rates in vasicek model
volRealEstate	Volatility
realEstate0	Initial value for real estate

**Examples**

```
data(ZC)
rRealEstate(horizon=5, nScenarios=10, ZC=ZC, vol=.1, k=2, volRealEstate=.15, realEstate0=50)
```

---

rShortRate

*rShortRate*


---

**Description**

Direct short rate generation. Object creation is managed internally.

**Usage**

```
rShortRate(horizon, nScenarios, ZC, vol, k)
```

**Arguments**

horizon	Horizon of projection
nScenarios	Number of scenarios
ZC	ZC rate input
vol	Volatility for short rates
k	k for rates in vasicek model

**Examples**

```
data(ZC)
rShortRate(horizon=15, nScenarios=10, ZC=ZC, vol=.1, k=2)
```

---

rStock

*rStock*


---

**Description**

Direct stock generation. Object creation is managed internally.

**Usage**

```
rStock(horizon, nScenarios, ZC, vol, k, volStock, stock0,
rho)
```

**Arguments**

horizon	Horizon of projection
nScenarios	Number of scenarios
ZC	ZC rate input
vol	Volatility for short rates
k	k for rates in vasicek model
volStock	Volatility
stock0	Initial value for stock
rho	Correlation between stock and short rates

**Examples**

```
data(ZC)
rStock(horizon=10, nScenarios=7, ZC=ZC, vol=.1, k=2, volStock=.2, stock0=100, rho=.5)
```

---

Scenarios	<i>Scenarios class</i>
-----------	------------------------

---

### Description

This is the main class of the package. It has several method to read and write the parameters.

### Details

**ParamsScenarios** A ParamsScenarios object containing the risk parameters

**ForwardRates** The forward rates

**ZCRates** Volatility for rates in vasicek model

**shortRatePaths** The short rate generated paths

**stockPaths** The stock generated paths

**realEstatePaths** The real estate generated paths

**liquiditySpreadPaths** The liquidity spread generated paths

**liquiditySpreadPaths** The liquidity spread generated paths

**defaultSpreadPaths** The default spread generated paths

---

setForwardRates	<i>setForwardRates method</i>
-----------------	-------------------------------

---

### Description

Calculate and set the forward rates in a Scenarios object. Internaly, this method uses the ForwardExtraction() function.

### Arguments

ZC	The Zero Coupon rates
horizon	Horizon for the calculation (in years)

### Examples

```
scenarios1 <- new("Scenarios")
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2,volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1,beta=1, eta=.05,rho=.5, stock0=100,realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
data(ZC)
scenarios1 <- setForwardRates(scenarios1, ZC, horizon=5)
```

---

```
setParamsBaseScenarios
      setParamsBaseScenarios method
```

---

**Description**

Set the horizon and nScenarios parameters of the [\[ParamsScenarios\]](#) sub-object of a Scenarios object

**Arguments**

horizon	Horizon for the projection (in years)
nScenarios	Number of scenarios

**Examples**

```
scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
```

---

```
setRiskParamsScenarios
      setRiskParamsScenarios method
```

---

**Description**

Set all the risk parameters of a Scenarios object (contained in a [\[ParamsScenarios\]](#) sub-object)

**Arguments**

vol	Volatility for rates in vasicek model
k	k for rates in vasicek model
volStock	Volatility for UL in Black & Scholes model
volRealEstate	Volatility for real estate in Black & Scholes model
volDefault	Volatility for LMN model
alpha	alpha for LMN model
beta	beta Volatility for LMN model
eta	eta Volatility for LMN model
rho	Correlation between stock and short rates
stock0	UL initial value
realEstate0	Real estate initial value
liquiditySpread0	Initial liquidity for LMN model
defaultSpread0	Initial default spread for LMN model

**Examples**

```

scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2, volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1, beta=1, eta=.05, rho=.5, stock0=100, realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)

```

---

setRiskParamsScenariosdefSpr

*setRiskParamsScenariosdefSpr method*


---

**Description**

Set risk parameters related to default spread in a Scenarios object (these parameters are contained in a [\[ParamsScenarios\]](#) sub-object)

**Arguments**

volDefault	Volatility for LMN model
defaultSpread0	Initial default spread for LMN model
alpha	alpha for LMN model
beta	beta Volatility for LMN model

---

setRiskParamsScenariosliqSpr

*setRiskParamsScenariosliqSpr method*


---

**Description**

Set risk parameters related to the spread in a Scenarios object (these parameters are contained in a [\[ParamsScenarios\]](#) sub-object)

**Arguments**

eta	eta Volatility for LMN model
liquiditySpread0	Initial liquidity for LMN model

---

 setRiskParamsScenariosRE

*setRiskParamsScenariosRE method*


---

**Description**

Set risk parameters related to real estates in a Scenarios object (these parameters are contained in a [\[ParamsScenarios\]](#) sub-object)

**Arguments**

vol	Volatility for rates in vasicek model
k	k for rates in vasicek model
volRealEstate	Volatility for real estate in Black & Scholes model
realEstate0	Real estate initial value

---

 setRiskParamsScenariosrt

*setRiskParamsScenariosrt method*


---

**Description**

Set risk parameters related to short rates in a Scenarios object (these parameters are contained in a [\[ParamsScenarios\]](#) sub-object)

**Arguments**

vol	Volatility for rates in vasicek model
k	k for rates in vasicek model

---

 setRiskParamsScenariosS

*setRiskParamsScenariosS method*


---

**Description**

Set risk parameters related to short rates in a Scenarios object (these parameters are contained in a [\[ParamsScenarios\]](#) sub-object)

**Arguments**

vol	Volatility for rates in vasicek model
k	k for rates in vasicek model
volStock	Volatility for UL in Black & Scholes model
stock0	UL initial value
rho	Correlation between stock and short rates

---

setZCRates	<i>setZCRates method</i>
------------	--------------------------

---

**Description**

Set the ZC rates in a Scenarios object. Internally, this method uses the ZCExtraction() function.

**Arguments**

ZC	The Zero Coupon rates
horizon	Horizon for the calculation (in years)

**Examples**

```

scenarios1 <- new("Scenarios")
scenarios1 <- setParamsBaseScenarios(scenarios1, horizon=5, nScenarios=10)
scenarios1 <- scenarios1 <- setRiskParamsScenarios(scenarios1, vol=.1, k=2, volStock=.2,
volRealEstate=.15, volDefault=.2, alpha=.1, beta=1, eta=.05, rho=.5, stock0=100, realEstate0=50,
liquiditySpread0=.01, defaultSpread0=.01)
data(ZC)
scenarios1 <- setZCRates(scenarios1, ZC, horizon=5)

```

---

ZC	<i>ZC data</i>
----	----------------

---

**Description**

ZC data for examples in the documentation

**Usage**

```
data(ZC)
```

**Examples**

```
data(ZC)
```

---

ZCBond\_PriceDistribution

*ZCBond\_PriceDistribution method*

---

**Description**

Get a distribution for ZC bond price.

**Arguments**

t	Date of pricing (has to be an integer)
T	Date of maturity

# Index

- \* **datasets**
  - ZC, [23](#)
- \* **package**
  - ESG-package, [2](#)
  
- Asset\_PriceDistribution, [3](#)
- Asset\_PriceDistribution, Scenarios-method  
(Asset\_PriceDistribution), [3](#)
  
- Bond\_PriceDistribution, [4](#)
- Bond\_PriceDistribution, Scenarios-method  
(Bond\_PriceDistribution), [4](#)
  
- CBond\_PriceDistribution, [4](#)
- CBond\_PriceDistribution, Scenarios-method  
(CBond\_PriceDistribution), [4](#)
- CDSPremium\_PriceDistribution, [5](#)
- CDSPremium\_PriceDistribution, Scenarios-method  
(CDSPremium\_PriceDistribution),  
[5](#)
- ConvBond\_PriceDistribution, [5](#)
- ConvBond\_PriceDistribution, Scenarios-method  
(ConvBond\_PriceDistribution), [5](#)
- customPathsGeneration, [6](#)
- customPathsGeneration, Scenarios-method  
(customPathsGeneration), [6](#)
  
- ESG (ESG-package), [2](#)
- ESG-package, [2](#)
- EuroCall\_Stock\_PriceDistribution, [6](#)
- EuroCall\_Stock\_PriceDistribution, Scenarios-method  
(EuroCall\_Stock\_PriceDistribution),  
[6](#)
- EuroCall\_ZC\_PriceDistribution, [7](#)
- EuroCall\_ZC\_PriceDistribution, Scenarios-method  
(EuroCall\_ZC\_PriceDistribution),  
[7](#)
- EuroPut\_Stock\_PriceDistribution, [7](#)
- EuroPut\_Stock\_PriceDistribution, Scenarios-method  
(EuroPut\_Stock\_PriceDistribution),  
[7](#)
- EuroPut\_ZC\_PriceDistribution, [7](#)
- EuroPut\_ZC\_PriceDistribution, Scenarios-method  
(EuroPut\_ZC\_PriceDistribution),  
[7](#)
  
- getdefaultSpreadPaths, [8](#)
- getdefaultSpreadPaths, Scenarios-method  
(getdefaultSpreadPaths), [8](#)
- getForwardRates, [8](#)
- getForwardRates, Scenarios-method  
(getForwardRates), [8](#)
- getLiquiditySpreadPaths, [8](#)
- getLiquiditySpreadPaths, Scenarios-method  
(getLiquiditySpreadPaths), [8](#)
- getParamsBaseScenarios, [8](#)
- getParamsBaseScenarios, Scenarios-method  
(getParamsBaseScenarios), [8](#)
- getrealEstatePaths, [9](#)
- getrealEstatePaths, Scenarios-method  
(getrealEstatePaths), [9](#)
- getRiskParamsScenarios, [9](#)
- getRiskParamsScenarios, Scenarios-method  
(getRiskParamsScenarios), [9](#)
- getRiskParamsScenariosdefSpr, [9](#)
- getRiskParamsScenariosdefSpr, Scenarios-method  
(getRiskParamsScenariosdefSpr),  
[9](#)
- getRiskParamsScenariosliqSpr, [10](#)
- getRiskParamsScenariosliqSpr, Scenarios-method  
(getRiskParamsScenariosliqSpr),  
[10](#)
- getRiskParamsScenariosRE, [10](#)
- getRiskParamsScenariosRE, Scenarios-method  
(getRiskParamsScenariosRE), [10](#)
- getRiskParamsScenariosrt, [10](#)
- getRiskParamsScenariosrt, Scenarios-method  
(getRiskParamsScenariosrt), [10](#)
- getRiskParamsScenariosS, [11](#)
- getRiskParamsScenariosS, Scenarios-method  
(getRiskParamsScenariosS), [11](#)

- getShortRatePaths, [11](#)
- getShortRatePaths, Scenarios-method  
(getShortRatePaths), [11](#)
- getstockPaths, [11](#)
- getstockPaths, Scenarios-method  
(getstockPaths), [11](#)
- getZCRates, [12](#)
- getZCRates, Scenarios-method  
(getZCRates), [12](#)
  
- MartingaleTest, [12](#)
- MartingaleTest, Scenarios-method  
(MartingaleTest), [12](#)
  
- ParamsScenarios, [13](#), [20–22](#)
- ParamsScenarios-class  
(ParamsScenarios), [13](#)
  
- rAllRisksFactors, [13](#)
- rAssetDistribution, [14](#)
- rDefaultSpread, [16](#)
- rLiquiditySpread, [16](#)
- rRealEstate, [17](#)
- rShortRate, [17](#)
- rStock, [18](#)
  
- Scenarios, [19](#)
- Scenarios-class (Scenarios), [19](#)
- setForwardRates, [19](#)
- setForwardRates, Scenarios-method  
(setForwardRates), [19](#)
- setParamsBaseScenarios, [20](#)
- setParamsBaseScenarios, Scenarios-method  
(setParamsBaseScenarios), [20](#)
- setRiskParamsScenarios, [20](#)
- setRiskParamsScenarios, Scenarios-method  
(setRiskParamsScenarios), [20](#)
- setRiskParamsScenariosdefSpr, [21](#)
- setRiskParamsScenariosdefSpr, Scenarios-method  
(setRiskParamsScenariosdefSpr),  
[21](#)
- setRiskParamsScenariosliqSpr, [21](#)
- setRiskParamsScenariosliqSpr, Scenarios-method  
(setRiskParamsScenariosliqSpr),  
[21](#)
- setRiskParamsScenariosRE, [22](#)
- setRiskParamsScenariosRE, Scenarios-method  
(setRiskParamsScenariosRE), [22](#)
- setRiskParamsScenariosrt, [22](#)
- setRiskParamsScenariosrt, Scenarios-method  
(setRiskParamsScenariosrt), [22](#)
- setRiskParamsScenariosS, [22](#)
- setRiskParamsScenariosS, Scenarios-method  
(setRiskParamsScenariosS), [22](#)
- setZCRates, [23](#)
- setZCRates, Scenarios-method  
(setZCRates), [23](#)
  
- ZC, [23](#)
- ZCBond\_PriceDistribution, [24](#)
- ZCBond\_PriceDistribution, Scenarios-method  
(ZCBond\_PriceDistribution), [24](#)