Package ‘LSMonteCarlo’

February 19, 2015

Type Package
Title American options pricing with Least Squares Monte Carlo method
Version 1.0
Date 2013-09-20
Author Mikhail A. Beketov
Maintainer Mikhail A. Beketov <mikhail.beketov@gmx.de>
Description The package compiles functions for calculating prices of American put options with Least Squares Monte Carlo method. The option types are plain vanilla American put, Asian American put, and Quanto American put. The pricing algorithms include variance reduction techniques such as Antithetic Variates and Control Variates. Additional functions are given to derive "price surfaces" at different volatilities and strikes, create 3-D plots, quickly generate Geometric Brownian motion, and calculate prices of European options with Black & Scholes analytical solution.
License GPL-3
Depends mvtnorm, fBasics, stats, utils, graphics, grDevices
NeedsCompilation no
Repository CRAN
Date/Publication 2013-09-23 23:07:43

R topics documented:

LSMonteCarlo-package .................................................. 2
AmerPutLSM ......................................................... 3
AmerPutLSMPriceSurf ............................................. 4
AmerPutLSM_AV ................................................... 6
AmerPutLSM.CV ................................................... 7
AsianAmerPutLSM ................................................ 9
AsianAmerPutLSMPriceSurf .................................... 10
EuPutBS ............................................................. 11
fastGBM ............................................................ 12
firstValueRow ..................................................... 13
price ................................................................. 13
American options pricing with Least Squares Monte Carlo method

Description

The package compiles functions that calculate prices of American put options with Least Squares Monte Carlo method. The option types are plain vanilla American put, Asian American put, and Quanto American put. The pricing algorithms include variance reduction techniques such as Antithetic Variates and Control Variates. Additional functions are given to derive "price surfaces" at different volatilities and strikes, create 3-D plots, quickly generate Geometric Brownian motion, and calculate prices of European options with Black & Scholes analytical solution.

Details

Package: LSMonteCarlo
Type: Package
Version: 1.0
Date: 2013-09-20
License: GPL 3

The Least Squares Monte Carlo is an approach developed to approximate the value of American options. It combines regression modeling and Monte Carlo simulation. The key feature of this method is estimation of the conditional expectation of the future pay-offs by a regression model (for details see Longstaff & Schwartz, 2000). The main pricing functions employing this method in the package are: AmerPutLSM, AsianAmerPutLSM, and QuantoAmerPutLSM. Pricing functions that include variance reduction methods are: AmerPutLSM_AV, QuantoAmerPutLSM_AV (Antithetic Variates) and AmerPutLSM_CV (Control Variates, with Black & Scholes solution for European put used as the control). All these functions are based on Geometric Brownian motion as a price process. They can be used with tailored summary, print, and price functions. The "price surfaces" at different volatilities and strikes can be derived using the functions AmerPutLSMPriceSurf, AsianAmerPutLSMPriceSurf, and QuantoAmerPutLSMPriceSurf, and plotted with tailored plot function. For general reading on option pricing with Monte Carlo methods see Glasserman (2004).

Author(s)

Mikhail A. Beketov
Maintainer: Mikhail A. Beketov <mikhail.beketov@gmx.de>
References

See Also
Functions: AmerPutLSM, AmerPutLSM_AV, AmerPutLSM_CV, AsianAmerPutLSM, QuantoAmerPutLSM, and QuantoAmerPutLSM_AV.

Examples
```r
Put<-AmerPutLSM(Spot=14.2, Strike=16.5, n=200, m=50)
summary(Put)
price(Put)
plot(AmerPutLSMPriceSurf(vols = (seq(0.1, 1.5, 0.2)), n=200, m=10,
strikes = (seq(0.5, 1.9, 0.2)), color = divPalette(150, "RdBu"))
```

AmerPutLSM Calculating the price of plain vanilla American put

Description
The function calculates the price of plain vanilla American put with Least Squares Monte Carlo method. The regression model included in the algorithm is quadratic polynomial (Longstaff & Schwartz, 2000).

Usage
```
AmerPutLSM(Spot = 1, sigma = 0.2, n = 1000, m = 365, Strike = 1.1, r = 0.06,
dr = 0, mT = 1)
```

## S3 method for class 'AmerPut'
print(x, ...)
## S3 method for class 'AmerPut'
summary(object, ...)

Arguments
- **Spot**: Spot price of the underlying asset (e.g. stock).
- **sigma**: Volatility of the underlying asset.
- **n**: Number of paths simulated.
- **m**: Number of time steps in the simulation.
- **Strike**: Strike price of the option.
- **r**: Interest rate of the numeraire currency (e.g. EUR).
Dividend rate of the underlying asset.

Maturity time (years).

An object returned by the functions AmerPutLSM.

An object returned by the function AmerPutLSM.

Not used.

The function returns an object of the class AmerPut that is a list comprising the price calculated, option type, and the entry parameters. Class-specific print function gives the option type information and the price. The price as a single number can be derived using the price function. An overview of the entire object can be seen using the summary function.

Mikhail A. Beketov


Functions: price, AmerPutLSM_AV, AmerPutLSM_CV, AsianAmerPutLSM, and QuantoAmerPutLSM.

Examples

amerputlsm()
put<-'amerputlsm(Spot=14.2, Strike=16.5, n=500, m=100)
put
summary(put)
price(put)
put$price

AmerPutLSMPriceSurf

Deriving a table of American put prices at different volatilities and strikes

The function calculates the prices at different volatilities and strikes using the AmerPutLSM function.
Usage

AmerPutLSMPriceSurf(Spot = 1, vols = seq(0.1, 2, 0.1), n = 1000, m = 365, strikess = seq(0.5, 2.5, 0.1), r = 0.06, dr = 0, mT = 1)

## S3 method for class 'PriceSurface'
summary(object, ...)
## S3 method for class 'PriceSurface'
plot(x, color = divPalette(800, "RdBu"), ...)

Arguments

- **Spot**: Spot price of the underlying asset (e.g. stock).
- **vols**: Sequence of volatilities.
- **n**: Number of paths simulated.
- **m**: Number of time steps in the simulation.
- **strikes**: Sequence of strikes.
- **r**: Interest rate of the numeraire currency (e.g. EUR).
- **dr**: Dividend rate of the underlying asset.
- **mT**: Maturity time (years).
- **object**: Object of the class PriceSurface that is a matrix of prices at different volatilities and strikes.
- **x**: Object of the class PriceSurface that is a matrix of prices at different volatilities and strikes.
- **color**: Color palette (the default pallet requires package fBasics, if you do not want to load this package, you can set color=NULL or other palette).
- **...**: Not used.

Value

The function returns an object of the class PriceSurface that is a matrix of prices at different volatilities and strikes. Class-specific summary function gives the sequences of volatilities and strikes used, as well as maximum, minimum, and average prices. Class-specific plot function constructs a 3-D plot of the price surface.

Author(s)

Mikhail A. Beketov

See Also

Examples

surface <- AmerPutLSM_PriceSurf(vols = (seq(0.1, 1.5, 0.2)), n=200, m=10,
strikes = (seq(0.5, 1.9, 0.2)))
summary(surface)
pplot(surface, color = divPalette(150, "RdBu"))

AmerPutLSM_AV  Pricing plain vanilla American put with Antithetic Variates

Description

The function calculates the price of a plain vanilla American put with Least Squares Monte Carlo method with Antithetic Variates (Glasserman, 2004). The regression model included in the algorithm is quadratic polynomial (Longstaff & Schwartz, 2000).

Usage

AmerPutLSM_AV(Spot = 1, sigma = 0.2, n = 1000, m = 365, Strike = 1.1, r = 0.06,
dr = 0, mT = 1)

## S3 method for class 'AmerPutAV'
print(x, ...)
## S3 method for class 'AmerPutAV'
summary(object, ...)

Arguments

Spot  Spot price of the underlying asset (e.g. stock).
sigma  Volatility of the underlying asset.
n  Number of paths simulated.
m  Number of time steps in the simulation.
Strike  Strike price of the option.
r  Interest rate of the numeraire currency (e.g. EUR).
dr  Dividend rate of the underlying asset.
mT  Maturity time (years).
x  An object returned by the functions AmerPutLSM_AV.
object  An object returned by the function AmerPutLSM_AV.
...  Not used.

Value

The function returns an object of the class AmerPutAV that is a list comprising the price calculated and the entry parameters. Class-specific print function gives the option type information and the price. The price as a single number can be derived using the price function. An overview of the entire object can be seen using the summary function.
Author(s)

Mikhail A. Beketov

References


See Also

Functions: price, AmerPutLSM, AmerPutLSM_AV, AsianAmerPutLSM, and QuantoAmerPutLSM.

Examples

AmerPutLSM_AV(n=500, m=50)
put<-AmerPutLSM_AV(Spot=14.2, Strike=16.5, n=200, m=50)
put
summary(put)
price(put)
put$price

---

### Pricing plain vanilla American put with Control Variates

**Description**

The function calculates the price of a plain vanilla American put with Least Squares Monte Carlo method with Control Variates (Glasserman, 2004). Black & Scholes solution for European put is used as the control. The regression model included in the algorithm is quadratic polynomial (Longstaff & Schwartz, 2000).

**Usage**

AmerPutLSM_AV(Spot = 1, sigma = 0.2, n = 1000, m = 365, Strike = 1.1, r = 0.06, dr = 0, mT = 1)

```r
## S3 method for class 'AmerPutCV'
print(x, ...)
## S3 method for class 'AmerPutCV'
summary(object, ...)
```
**Arguments**

- **Spot**: Spot price of the underlying asset (e.g. stock).
- **sigma**: Volatility of the underlying asset.
- **n**: Number of paths simulated.
- **m**: Number of time steps in the simulation.
- **Strike**: Strike price of the option.
- **r**: Interest rate of the numeraire currency (e.g. EUR).
- **dr**: Dividend rate of the underlying asset.
- **mT**: Maturity time (years).
- **x**: An object returned by the functions AmerPutLSM.CV.
- **object**: An object returned by the function AmerPutLSM.CV.
- **...**: Not used.

**Value**

The function returns an object of the class AmerPutCV that is a list comprising the price calculated and the entry parameters. Class-specific `print` function gives the option type information and the price. The price as a single number can be derived using the `price` function. An overview of the entire object can be seen using the `summary` function.

**Author(s)**

Mikhail A. Beketov

**References**


**See Also**


**Examples**

```r
AmerPutLSM.CV()  
put<-AmerPutLSM.CV(Spot=14.2, Strike=16.5, n=200, m=50)  
put
summary(put)
price(put)
put$price
```
Calculating the price of Asian American put

Description

The function calculates the price of Asian American put with Least Squares Monte Carlo method (pay-off based on arithmetic mean). The regression model included in the algorithm is quadratic polynomial (Longstaff & Schwartz, 2000).

Usage

\[
\text{AsianAmerPutLSM} (\text{Spot} = 1, \text{sigma} = 0.2, n = 1000, m = 365, \text{Strike} = 1.1, r = 0.06, \text{dr} = 0, \text{mT} = 1)
\]

## S3 method for class 'AsianAmerPut'

\[
\text{print(x, \ldots)}
\]

## S3 method for class 'AsianAmerPut'

\[
\text{summary(object, \ldots)}
\]

Arguments

- **Spot**: Spot price of the underlying asset (e.g. stock).
- **sigma**: Volatility of the underlying asset.
- **n**: Number of paths simulated.
- **m**: Number of time steps in the simulation.
- **Strike**: Strike price of the option.
- **r**: Interest rate of the numeraire currency (e.g. EUR).
- **dr**: Dividend rate of the underlying asset.
- **mT**: Maturity time (years).
- **x**: An object returned by the functions AsianAmerPutLSM.
- **object**: An object returned by the function AsianAmerPutLSM.
- \ldots: Not used.

Value

The function returns an object of the class AsianAmerPut that is a list comprising the price calculated, option type, and the entry parameters. Class-specific print function gives the option type information and the price. The price as a single number can be derived using the price function. An overview of the entire object can be seen using the summary function.

Author(s)

Mikhail A. Beketov
References


See Also

Functions: price, AmerPutLSM, AmerPutLSM_CV, AmerPutLSM_AV, and QuantoAmerPutLSM.

Examples

```r
asianamerputlsm(n=UPPL m=QPP)
pot<-asianamerputlsm(Spot=14.2, Strike=16.5, n=500, m=50)
pot
summary(pot)
price(pot)
pot$price
```

---

Deriving a table of Asian American put prices at different volatilities and strikes

Description

The function calculates the prices at different volatilities and strikes using the asianamerputlsm function.

Usage

```r
asianamerputlsmpricesurf(Spot = 1, vols = (seq(0.1, 2, 0.1)), n = 1000, m = 365,
strikes = (seq(0.5, 2.5, 0.1)), r = 0.06, dr = 0, mT = 1)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot</td>
<td>Spot price of the underlying asset (e.g. stock).</td>
</tr>
<tr>
<td>vols</td>
<td>Sequence of volatilities.</td>
</tr>
<tr>
<td>n</td>
<td>Number of paths simulated.</td>
</tr>
<tr>
<td>m</td>
<td>Number of time steps in the simulation.</td>
</tr>
<tr>
<td>strikes</td>
<td>Sequence of strikes.</td>
</tr>
<tr>
<td>r</td>
<td>Interest rate of the numeraire currency (e.g. EUR).</td>
</tr>
<tr>
<td>dr</td>
<td>Dividend rate of the underlying asset.</td>
</tr>
<tr>
<td>mT</td>
<td>Maturity time (years).</td>
</tr>
</tbody>
</table>
The function returns an object of the class PriceSurface that is a matrix of prices at different volatilities and strikes. Class-specific summary function gives the sequences of volatilities and strikes used, as well as maximum, minimum, and average prices. Class-specific plot function constructs a 3-D plot of the price surface.

Author(s)
Mikhail A. Beketov

See Also

Examples
surface <- AsianAmerPutLSMPriceSurf(vols = (seq(0.1, 1.5, 0.2)), n=200, m=10, strikes = (seq(0.5, 1.9, 0.2)))
summary(surface)
plot(surface, color = divPalette(150, "RdBu"))

EuPutBS

Black & Scholes solution for European put and call

Description
Pricing plain vanilla American put and call options using Black & Scholes solution.

Usage
EuPutBS(Spot, sigma, Strike, r, dr, mT)
EuCallBS(Spot, sigma, Strike, r, dr, mT)

Arguments
Spot Spot price of the underlying asset (e.g. stock).
sigma Volatility of the underlying asset.
Strike Strike price of the option.
r Interest rate of the numeraire currency (e.g. EUR).
Dr Dividend rate of the underlying asset.
mT Maturity time (years).

Value
The function returns the price as a single number (class "numeric").
See Also

AmerPutLSM.CV

Examples

EuPutBS(1, 0.2, 1, 0.06, 0, 1)
EuCallBS(1, 0.2, 1, 0.06, 0, 1)

Description

Quick Generating Geometric Brownian motion avoiding unnecessary loops using the cumsum function. Technical function implemented in the pricing functions of the package.

Usage

fastGBM(Spot = 1, sigma = 0.2, n = 1000, m = 365, r = 0.06, dr = 0, mT = 1)

Arguments

Spot            Spot price of the underlying asset (e.g. stock).
sigma           Volatility of the underlying asset.
n               Number of paths simulated.
m               Number of time steps in the simulation.
r               Interest rate of the numeraire currency (e.g. EUR).
dr              Dividend rate of the underlying asset.
mT             Maturity time (years).

Value

Table with paths generated (each row is a path, class "matrix")

Author(s)

Mikhail A. Beketov

See Also

Functions: AmerPutLSM, AmerPutLSM_AV, AmerPutLSM.CV, AsianAmerPutLSM, QuantoAmerPutLSM, and QuantoAmerPutLSM_AV.

Examples

fastGBM(n=10, m=5)
matplot(t(fastGBM(n=100, m=100)), type="l") # matrix transpose by "t()"
**firstValueRow**

*Returning the first >0 value in each row of a matrix*

**Description**

Technical function implemented in the pricing functions of the package. It returns the first >0 value in each row of a matrix and assign zero to all subsequent values.

**Usage**

`firstValueRow(x)`

**Arguments**

- `x` A matrix.

**Value**

A matrix.

**Author(s)**

Mikhail A. Beketov

**See Also**


**Examples**

```r
mat <- matrix(c(0,0,2,0,4,0,3,0,1,9,8,7), ncol=4)
mat
firstValueRow(mat)
```

---

**price**

*Extracting price from the pricing functions outputs*

**Description**

The function is nothing else, but the object$price action, with the object returned by the pricing functions in the package.

**Usage**

`price(x)`
Arguments

x  Object returned by the pricing functions in the package

Value

The function returns the price as a single number (class "numeric").

See Also

Functions: AmerPutLSM, AmerPutLSM_AV, AmerPutLSM_CV, AsianAmerPutLSM, QuantoAmerPutLSM, and QuantoAmerPutLSM_AV.

Examples

```r
put <- AmerPutLSM()
price(put)
put$price
```

---

**Description**

The function calculates the price of Quanto American put with Least Squares Monte Carlo method. The Quanto option is cash-settled option, whose pay-off is converted into a third currency/asset at exercise at a pre-specified rate/price (Wystup, 2011), and can also be considered as a usual option but settled in a "wrong" asset (Vecer, 2011). The regression model included in the algorithm is quadratic polynomial (Longstaff & Schwartz, 2000).

**Usage**

```r
QuantoAmerPutLSM(Spot = 1, sigma = 0.2, n = 1000, m = 365, Strike = 1.1, r = 0.06, 
dr = 0, mT = 1, Spot2 = 1, sigma2 = 0.2, r2 = 0, dr2 = 0, rho = 0)
```

```r
## S3 method for class 'QuantoAmerPut'
print(x, ...) 
## S3 method for class 'QuantoAmerPut'
summary(object, ...)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot</td>
<td>Spot price of the underlying asset (e.g. stock).</td>
</tr>
<tr>
<td>sigma</td>
<td>Volatility of the underlying asset.</td>
</tr>
<tr>
<td>n</td>
<td>Number of paths simulated.</td>
</tr>
<tr>
<td>m</td>
<td>Number of time steps in the simulation.</td>
</tr>
</tbody>
</table>
**QuantoAmerPutLSM**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strike</td>
<td>Strike price of the option.</td>
</tr>
<tr>
<td>r</td>
<td>Interest rate of the numeraire currency (e.g. USD).</td>
</tr>
<tr>
<td>dr</td>
<td>Dividend rate of the underlying asset.</td>
</tr>
<tr>
<td>mT</td>
<td>Maturity time (years).</td>
</tr>
<tr>
<td>Spot2</td>
<td>Spot price of the 3rd asset (e.g. EUR/USD).</td>
</tr>
<tr>
<td>sigma2</td>
<td>Volatility of the 3rd asset.</td>
</tr>
<tr>
<td>r2</td>
<td>Interest rate of the 3rd asset.</td>
</tr>
<tr>
<td>dr2</td>
<td>Dividend rate of the 3rd asset.</td>
</tr>
<tr>
<td>rho</td>
<td>Correlation coefficient between the prices.</td>
</tr>
<tr>
<td>x</td>
<td>An object returned by the functions QuantoAmerPutLSM.</td>
</tr>
<tr>
<td>object</td>
<td>An object returned by the function QuantoAmerPutLSM.</td>
</tr>
<tr>
<td>...</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

**Value**

The function returns an object of the class QuantoAmerPut that is a list comprising the price calculated, option type, and the entry parameters. Class-specific print function gives the option type information and the price. The price as a single number can be derived using the price function. An overview of the entire object can be seen using the summary function.

**Note**

The function rmvnorm included in the pricing algorithm is a part of the mnormt package. Please, load that package before the use of the QuantoAmerPutLSM function.

**Author(s)**

Mikhail A. Beketov

**References**


**See Also**

Functions: price, QuantoAmerPutLSM_AV, AmerPutLSM, AsianAmerPutLSM, and AmerPutLSM_AV.

**Examples**

```r
QuantoAmerPutLSM(n=200, m=50)
put<QuantoAmerPutLSM(Spot=14.2, Strike=16.5, n=200, m=50)
put
summary(put)
price(put)
```
Deriving a table of Quanto American put prices at different volatilities and strikes

Description

The function calculates the prices at different volatilities and strikes using the QuantoAmerPutLSM function.

Usage

QuantoAmerPutLSMPriceSurf(Spot = 1, vols = (seq(0.1, 2, 0.1)), n = 1000, m = 365, strikes = (seq(0.5, 2.5, 0.1)), r = 0.06, dr = 0, mT = 1, Spot2 = 1, sigma2 = 0.2, r2 = 0, dr2 = 0, rho = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot</td>
<td>Spot price of the underlying asset (e.g. stock).</td>
</tr>
<tr>
<td>vols</td>
<td>Sequence of volatilities.</td>
</tr>
<tr>
<td>n</td>
<td>Number of paths simulated.</td>
</tr>
<tr>
<td>m</td>
<td>Number of time steps in the simulation.</td>
</tr>
<tr>
<td>strikes</td>
<td>Sequence of strikes.</td>
</tr>
<tr>
<td>r</td>
<td>Interest rate of the numeraire currency (e.g. USD).</td>
</tr>
<tr>
<td>dr</td>
<td>Dividend rate of the underlying asset.</td>
</tr>
<tr>
<td>mT</td>
<td>Maturity time (years).</td>
</tr>
<tr>
<td>Spot2</td>
<td>Spot price of the 3rd asset (e.g. EUR/USD).</td>
</tr>
<tr>
<td>sigma2</td>
<td>Volatility of the 3rd asset.</td>
</tr>
<tr>
<td>r2</td>
<td>Interest rate of the 3rd asset.</td>
</tr>
<tr>
<td>dr2</td>
<td>Dividend rate of the 3rd asset.</td>
</tr>
<tr>
<td>rho</td>
<td>Correlation coefficient between the prices.</td>
</tr>
</tbody>
</table>

Value

The function returns an object of the class PriceSurface that is a matrix of prices at different volatilities and strikes. Class-specific summary function gives the sequences of volatilities and strikes used, as well as maximum, minimum, and average prices. Class-specific plot function constructs a 3-D plot of the price surface.

Note

The function rmvnorm included in the pricing algorithm is a part of the mnormt package. Please, load that package before the use of the QuantoAmerPutLSMPriceSurf function. Using the function plot with default pallet requires package fBasics, if you do not want to load this package, you can set color=NULL or other palette).
QuantoAmerPutLSM_AV

Author(s)

Mikhail A. Beketov

See Also


Examples

surface <- QuantoAmerPutLSMPriceSurf(vols = (seq(0.1, 1.7, 0.2)), n=100, m=5, strikes = (seq(0.7, 1.7, 0.2)))
summary(surface)
plot(surface, color = divPalette(150, "RdBu"))

Pricing Quanto American put with Antithetic Variates

Description

The function calculates the price of Quanto American put with Least Squares Monte Carlo method with Antithetic Variates (Glasserman, 2004). The Quanto option is cash-settled option, whose pay-off is converted into a third currency/asset at exercise at a pre-specified rate/price (Wystup, 2011), and can also be considered as a usual option but settled in a "wrong" asset (Vecer, 2011). The regression model included in the algorithm is quadratic polynomial (Longstaff & Schwartz, 2000).

Usage

QuantoAmerPutLSM_AV(Spot = 1, sigma = 0.2, n = 1000, m = 365, Strike = 1.1, r = 0.06, dr = 0, mT = 1, Spot2 = 1, sigma2 = 0.2, r2 = 0, dr2 = 0, rho = 0)

## S3 method for class 'QuantoAmerPut_AV'
print(x, ...
## S3 method for class 'QuantoAmerPut_AV'
summary(object, ...)
dr  Dividend rate of the underlying asset.
mT  Maturity time (years).
Spot2  Spot price of the 3rd asset (e.g. EUR/USD).
sigma2  Volatility of the 3rd asset.
r2  Interest rate of the 3rd asset.
dr2  Dividend rate of the 3rd asset.
rho  Correlation coefficient between the prices.
x  An object returned by the functions QuantoAmerPutLSM_AV.
object  An object returned by the function QuantoAmerPutLSM_AV.
...  Not used.

Value

The function returns an object of the class QuantoAmerPut_AV that is a list comprising the price calculated, option type, and the entry parameters. Class-specific print function gives the option type information and the price. The price as a single number can be derived using the price function. An overview of the entire object can be seen using the summary function.

Note

The function rmvnorm included in the pricing algorithm is a part of the mnormt package. Please, load that package before the use of the QuantoAmerPutLSM_AV function.

Author(s)

Mikhail A. Beketov

References


See Also

Functions: price, QuantoAmerPutLSM, AmerPutLSM, AsianAmerPutLSM, and AmerPutLSM_AV.

Examples

QuantoAmerPutLSM_AV(n=200, m=50)
put<-QuantoAmerPutLSM_AV(Spot=14.2, Strike=16.5, n=200, m=50)
put
summary(put)
price(put)
Index

*Topic American call
EuPutBS, 11

*Topic American put
AmerPutLSM, 3
AmerPutLSM_AV, 6
AmerPutLSM_CV, 7
AmerPutLSMPriceSurf, 4
AsianAmerPutLSM, 9
AsianAmerPutLSMPriceSurf, 10
EuPutBS, 11
fastGBM, 12
firstValueRow, 13
price, 13
QuantoAmerPutLSM, 14
QuantoAmerPutLSM_AV, 17
QuantoAmerPutLSMPriceSurf, 16

*Topic Monte Carlo
AmerPutLSM, 3
AmerPutLSM_AV, 6
AmerPutLSM_CV, 7
AmerPutLSMPriceSurf, 4
AsianAmerPutLSM, 9
AsianAmerPutLSMPriceSurf, 10
fastGBM, 12
firstValueRow, 13
LSMonteCarlo-package, 2
price, 13
QuantoAmerPutLSM, 14
QuantoAmerPutLSM_AV, 17
QuantoAmerPutLSMPriceSurf, 16

*Topic Quantitative Finance
LSMonteCarlo-package, 2

*Topic Regression
LSMonteCarlo-package, 2

*Topic Option pricing
AmerPutLSM, 3
AmerPutLSM_AV, 6
AmerPutLSM_CV, 7
AmerPutLSMPriceSurf, 4
AsianAmerPutLSM, 9
AsianAmerPutLSMPriceSurf, 10
EuPutBS, 11
fastGBM, 12
firstValueRow, 13
LSMonteCarlo-package, 2
price, 13
QuantoAmerPutLSM, 14
QuantoAmerPutLSM_AV, 17
QuantoAmerPutLSMPriceSurf, 16

plot.PriceSurface, 11, 17
plot.PriceSurface(#AmerPutLSMPriceSurf), 4
price, 4, 7, 8, 10, 13, 15, 18
print.AmerPut(AmerPutLSM), 3
print.AmerPutAV(AmerPutLSM_AV), 6
print.AmerPutCV(AmerPutLSM_CV), 7
print.AsianAmerPut(AsianAmerPutLSM), 9
print.QuantoAmerPut(QuantoAmerPutLSM), 14
print.QuantoAmerPut_AV(#QuantoAmerPutLSM_AV), 17
QuantoAmerPutLSM, 3, 4, 7, 8, 10, 12–14, 14, 17, 18
QuantoAmerPutLSM_AV, 3, 12–15, 17
QuantoAmerPutLSMPriceSurf, 5, 11, 16

summary.AmerPut (AmerPutLSM), 3
summary.AmerPutAV (AmerPutLSM_AV), 6
summary.AmerPutCV (AmerPutLSM_CV), 7
summary.AsianAmerPut (AsianAmerPutLSM), 9
summary.PriceSurface, 11, 17
summary.PriceSurface
   (AmerPutLSMPriceSurf), 4
summary.QuantoAmerPut
   (QuantoAmerPutLSM), 14
summary.QuantoAmerPut_AV
   (QuantoAmerPutLSM_AV), 17