

Package ‘greeks’

January 31, 2024

Title Sensitivities of Prices of Financial Options and Implied Volatilities

Version 1.3.2

Description Methods to calculate sensitivities of financial option prices for European, geometric and arithmetic Asian, and American options, with various payoff functions in the Black Scholes model, and in more general jump diffusion models. A shiny app to interactively plot the results is included. Furthermore, methods to compute implied volatilities are provided for a wide range of option types and custom payoff functions. Classical formulas are implemented for European options in the Black Scholes Model, as is presented in Hull, J. C. (2017), Options, Futures, and Other Derivatives. In the case of Asian options, Malliavin Monte Carlo Greeks are implemented, see Hudde, A. & Rüschendorf, L. (2023). European and Asian Greeks for exponential Lévy processes. <doi:10.1007/s11009-023-10014-5>. For American options, the Binomial Tree Method is implemented, as is presented in Hull, J. C. (2017).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), R.rsp

Config/testthat/edition 3

Imports magrittr, dqrng, Rcpp, tibble, ggplot2, plotly, shiny, tidyr

LinkingTo Rcpp

NeedsCompilation yes

Author Anselm Hudde [aut, cre] (<<https://orcid.org/0000-0002-5652-2815>>)

Maintainer Anselm Hudde <anselmhudde@gmx.de>

VignetteBuilder knitr

Repository CRAN

Date/Publication 2024-01-31 22:40:02 UTC

R topics documented:

Binomial_American_Greeks	2
BS_European_Greeks	3
BS_Geometric_Asian_Greeks	4
BS_Implied_Volatility	5
BS_Malliavin_Asian_Greeks	6
Greeks	7
Greeks_UI	9
Implied_Volatility	9
Malliavin_Asian_Greeks	10
Malliavin_European_Greeks	12
Malliavin_Geometric_Asian_Greeks	13

Index	15
--------------	-----------

Binomial_American_Greeks

Computes the Greeks of an American call- or put-option with the Binomial options pricing model

Description

Computes the Greeks of an American call- or put-option with the Binomial options pricing model

Usage

```
Binomial_American_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  dividend_yield = 0,
  payoff = "call",
  greek = c("fair_value", "delta", "vega", "theta", "rho", "epsilon", "gamma"),
  steps = 1000,
  eps = 1/1e+05
)
```

Arguments

initial_price	• initial price of the underlying asset.
exercise_price	• strike price of the option.
r	• risk-free interest rate.
time_to_maturity	• time to maturity.

volatility	• volatility of the underlying asset.
dividend_yield	• dividend yield.
payoff	• the payoff function, a string in ("call", "put").
greek	• the Greek to be calculated.
steps	• the number of integration steps.
eps	• the step size for the finite difference method to calculate theta, vega, rho and epsilon

Value

Named vector containing the values of the Greeks specified in the parameter greek.

Examples

```
Binomial_American_Greeks(initial_price = 100, exercise_price = 100,
  r = 0, time_to_maturity = 1, volatility = 0.3, dividend_yield = 0,
  payoff = "call", greek = c("fair_value", "delta", "vega", "theta", "rho",
  "epsilon", "gamma"), steps = 20)
```

BS_European_Greeks	<i>Computes the Greeks of a European call- or put-option, or of digital options in the Black Scholes model</i>
--------------------	--

Description

Computes the Greeks of a European call- or put-option, or of digital options in the Black Scholes model

Usage

```
BS_European_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  dividend_yield = 0,
  payoff = "call",
  greek = c("fair_value", "delta", "vega", "theta", "rho", "epsilon", "lambda", "gamma",
  "vanna", "charm", "vomma", "veta", "speed")
)
```

Arguments

initial_price	• initial price of the underlying asset
exercise_price	• strike price of the option
r	• risk-free interest rate
time_to_maturity	• time to maturity in years
volatility	• volatility of the underlying asset
dividend_yield	• dividend yield
payoff	• in c("call", "put", "cash_or_nothing_call", "cash_or_nothing_put", "asset_or_nothing_call", "asset_or_nothing_put")
greek	• Greeks to be calculated in c("fair_value", "delta", "vega", "theta", "rho", "epsilon", "lambda", "gamma", "vanna", "charm", "vomma", "veta", "vera", "speed", "zomma", "color", "ultima")

Value

Named vector containing the values of the Greeks specified in the parameter greek.

Examples

```
BS_European_Greeks(initial_price = 120, exercise_price = 100,
  r = 0.02, time_to_maturity = 4.5, dividend_yield = 0.015, volatility = 0.22,
  greek = c("fair_value", "delta", "gamma"), payoff = "put")
```

BS_Geometric_Asian_Greeks

Computes the Greeks of a Geometric Asian Option with classical Call- and Put-Payoff in the Black Scholes model

Description

Computes the Greeks of a Geometric Asian Option with classical Call- and Put-Payoff in the Black Scholes model

Usage

```
BS_Geometric_Asian_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  dividend_yield = 0,
  payoff = "call",
  greek = c("fair_value", "delta", "rho", "vega", "theta", "gamma")
)
```

Arguments

<code>initial_price</code>	• initial price of the underlying asset, can also be a vector
<code>exercise_price</code>	• strike price of the option
<code>r</code>	• risk-free interest rate
<code>time_to_maturity</code>	• time to maturity in years
<code>volatility</code>	• volatility of the underlying asset
<code>dividend_yield</code>	• dividend yield
<code>payoff</code>	• the payoff function, either a string in ("call", "put")
<code>greek</code>	• the Greeks to be calculated in c("fair_value", "delta", "vega", "theta", "rho", "gamma", "vomma")

Value

Named vector containing the values of the Greeks specified in the parameter `greek`.

Examples

```
BS_Geometric_Asian_Greeks(initial_price = 110, exercise_price = 100,
r = 0.02, time_to_maturity = 4.5, dividend_yield = 0.015, volatility = 0.22,
greek = c("fair_value", "delta", "rho", "vega", "theta", "gamma"),
payoff = "put")
```

`BS_Implied_Volatility` *Computes the implied volatility for European options in the Black Scholes model via Halley's method.*

Description

Computes the implied volatility for European options in the Black Scholes model via Halley's method.

Usage

```
BS_Implied_Volatility(
  option_price,
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  dividend_yield = 0,
  payoff = "call",
  start_volatility = 0.3,
  precision = 1e-09
)
```

Arguments

option_price	• current price of the option
initial_price	• initial price of the underlying asset.
exercise_price	• strike price of the option.
r	• risk-free interest rate.
time_to_maturity	• time to maturity.
dividend_yield	• dividend yield.
payoff	• the payoff function, a string in ("call", "put").
start_volatility	• the volatility value to start the approximation
precision	• precision of the result

Value

Named vector containing the values of the Greeks specified in the parameter greek.

Examples

```
BS_Implied_Volatility(option_price = 27, initial_price = 100,
exercise_price = 100, r = 0.03, time_to_maturity = 5, dividend_yield = 0.015,
payoff = "call")
```

BS_Malliavin_Asian_Greeks

Computes the Greeks of an Asian option with the Malliavin Monte Carlo Method in the Black Scholes model

Description

Computes the Greeks of an Asian option with the Malliavin Monte Carlo Method in the Black Scholes model

Usage

```
BS_Malliavin_Asian_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  dividend_yield = 0,
  payoff = "call",
  greek = c("fair_value", "delta", "vega", "rho"),
  steps = round(time_to_maturity * 252),
  paths = 1000,
  seed = 1
)
```

Arguments

- initial_price • initial price of the underlying asset, can also be a vector
- exercise_price • strike price of the option, can also be a vector
- r • risk-free interest rate
- time_to_maturity • time to maturity in years
- volatility • volatility of the underlying asset
- dividend_yield • dividend yield
- payoff • the payoff function, either a string in ("call", "put"), or a function
- greek • Greeks to be calculated in c("fair_value", "delta", "gamma", "rho", "theta", "vega")
- steps • the number of integration steps
- paths • the number of simulated paths
- seed • the seed of the random number generator

Value

Named vector containing the values of the Greeks specified in the parameter greek.

Examples

```
BS_Malliavin_Asian_Greeks(initial_price = 110, exercise_price = 100,
r = 0.02, time_to_maturity = 4.5, dividend_yield = 0.015, volatility = 0.22,
greek = c("fair_value", "delta", "rho"), payoff = "put")
```

Greeks	<i>Computes the Greeks of various options in the Black Scholes model or both in the Black Scholes model or a Jump Diffusion model in the case of Asian Options, or in the Binomial options pricing model</i>
--------	--

Description

Computes the Greeks of various options in the Black Scholes model or both in the Black Scholes model or a Jump Diffusion model in the case of Asian Options, or in the Binomial options pricing model

Usage

```
Greeks(
  initial_price,
  exercise_price,
  r,
  time_to_maturity,
```

```

    volatility,
    dividend_yield = 0,
    model = "Black_Scholes",
    option_type = "European",
    payoff = "call",
    greek = c("fair_value", "delta", "vega", "theta", "rho", "gamma"),
    antithetic = TRUE,
    ...
)

```

Arguments

<code>initial_price</code>	• initial price of the underlying asset
<code>exercise_price</code>	• strike price of the option
<code>r</code>	• risk-free interest rate
<code>time_to_maturity</code>	• time to maturity in years
<code>volatility</code>	• volatility of the underlying asset
<code>dividend_yield</code>	• dividend yield
<code>model</code>	• the model to be chosen in ("black_scholes", "jump_diffusion")
<code>option_type</code>	in c("European", "American", "Asian", "Geometric Asian", "Digital", "Binomial") - the type of option to be considered
<code>payoff</code>	• in c("call", "put", "cash_or_nothing_call", "cash_or_nothing_put", "asset_or_nothing_call", "asset_or_nothing_put")
<code>greek</code>	• Greeks to be calculated in c("fair_value", "delta", "vega", "theta", "rho", "epsilon", "lambda", "gamma", "vanna", "charm", "vomma", "veta", "vera", "speed", "zomma", "color", "ultima")
<code>antithetic</code>	• if TRUE, antithetic random numbers will be chosen to decrease variance
<code>...</code>	• ... Other arguments passed on to methods

Value

Named vector containing the values of the Greeks specified in the parameter `greek`.

Examples

```
Greeks(initial_price = 100, exercise_price = 120, r = 0.01,
time_to_maturity = 5, volatility = 0.30, payoff = "call")
```

```
Greeks(initial_price = 100, exercise_price = 100, r = -0.005,
time_to_maturity = 1, volatility = 0.30, payoff = "put",
option_type = "American")
```

Greeks_UI	<i>Opens a shiny app to plot option prices and Greeks</i>
-----------	---

Description

Opens a shiny app to plot option prices and Greeks

Usage

Greeks_UI()

Implied_Volatility	<i>Computes the implied volatility for various options via Newton's method</i>
--------------------	--

Description

Computes the implied volatility for various options via Newton's method

Usage

```
Implied_Volatility(
  option_price,
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  dividend_yield = 0,
  model = "Black_Scholes",
  option_type = "European",
  payoff = "call",
  start_volatility = 0.3,
  precision = 1e-06,
  max_iter = 30
)
```

Arguments

option_price	• current price of the option
initial_price	• initial price of the underlying asset
exercise_price	• strike price of the option
r	• risk-free interest rate
time_to_maturity	• time to maturity in years

dividend_yield	• dividend yield
model	• the model to be chosen
option_type	in c("European", "American", "Geometric Asian", "Asian", "Digital") - the type of option to be considered
payoff	• in c("call", "put")
start_volatility	initial guess
precision	precision of the computation
max_iter	maximal number of iterations of the approximation

Value

Named vector containing the values of the greeks specified in the parameter greek.

Examples

```
Implied_Volatility(15, r = 0.05, option_type = "Asian",
  payoff = "call")
```

Malliavin_Asian_Greeks

Computes the Greeks of an Asian option with the Malliavin Monte Carlo Method in the Black Scholes model, or for Asian options, also in a Jump Diffusion model

Description

Computes the Greeks of an Asian option with the Malliavin Monte Carlo Method in the Black Scholes model, or for Asian options, also in a Jump Diffusion model

Usage

```
Malliavin_Asian_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  dividend_yield = 0,
  payoff = "call",
  greek = c("fair_value", "delta", "rho", "vega", "theta", "gamma"),
  model = "black_scholes",
  lambda = 0.2,
  alpha = 0.3,
  jump_distribution = function(n) stats::rt(n, df = 3),
  steps = round(time_to_maturity * 252),
```

```

    paths = 10000,
    seed = 1,
    antithetic = FALSE
  )

```

Arguments

<code>initial_price</code>	• initial price of the underlying asset, can also be a vector
<code>exercise_price</code>	• strike price of the option, can also be a vector
<code>r</code>	• risk-free interest rate
<code>time_to_maturity</code>	• time to maturity in years
<code>volatility</code>	• volatility of the underlying asset
<code>dividend_yield</code>	• dividend yield
<code>payoff</code>	• the payoff function, either a string in ("call", "put", "digital_call", "digital_put"), or a function
<code>greek</code>	• the Greek to be calculated
<code>model</code>	• the model to be chosen in ("black_scholes", "jump_diffusion")
<code>lambda</code>	• the lambda of the Poisson process in the jump-diffusion model
<code>alpha</code>	• the alpha in the jump-diffusion model influences the jump size
<code>jump_distribution</code>	• the distribution of the jumps, choose a function which generates random numbers with the desired distribution
<code>steps</code>	• the number of integration steps
<code>paths</code>	• the number of simulated paths
<code>seed</code>	• the seed of the random number generator
<code>antithetic</code>	• if TRUE, antithetic random numbers will be chosen to decrease variance

Value

Named vector containing the values of the Greeks specified in the parameter `greek`.

Examples

```

Malliavin_Asian_Greeks(initial_price = 110, exercise_price = 100,
  r = 0.02, time_to_maturity = 4.5, dividend_yield = 0.015, volatility = 0.22,
  greek = c("fair_value", "delta", "rho"), payoff = "put")

```

Malliavin_European_Greeks

Computes the Greeks of a European option with the Malliavin Monte Carlo Method in the Black Scholes model

Description

Computes the Greeks of a European option with the Malliavin Monte Carlo Method in the Black Scholes model

Usage

```
Malliavin_European_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  payoff = "call",
  greek = c("fair_value", "delta", "vega", "theta", "rho", "gamma"),
  model = "Black Scholes",
  paths = 10000,
  seed = 1,
  antithetic = FALSE
)
```

Arguments

initial_price	• initial price of the underlying asset
exercise_price	• strike price of the option
r	• risk-free interest rate
time_to_maturity	• time to maturity in years
volatility	• volatility of the underlying asset
payoff	• the payoff function, either a string in ("call", "put", "cash_or_nothing_call", "cash_or_nothing_call", "asset_or_nothing_call", "asset_or_nothing_put"), or a function
greek	• the Greeks to be calculated in ("fair_value", "delta", "vega", "theta", "rho", "gamma")
model	• the model to be chosen
paths	• the number of simulated paths
seed	• the seed of the random number generator
antithetic	• if TRUE, antithetic random numbers will be chosen to decrease variance

Value

Named vector containing the values of the Greeks specified in the parameter greek

Examples

```
Malliavin_European_Greeks(initial_price = 110,
exercise_price = 100, r = 0.02, time_to_maturity = 4.5,
volatility = 0.22, greek = c("fair_value", "delta", "rho"), payoff = "put")
```

Malliavin_Geometric_Asian_Greeks

Computes the Greeks of an Asian option with the Malliavin Monte Carlo Method in the Black Scholes model

Description

Computes the Greeks of an Asian option with the Malliavin Monte Carlo Method in the Black Scholes model

Usage

```
Malliavin_Geometric_Asian_Greeks(
  initial_price = 100,
  exercise_price = 100,
  r = 0,
  time_to_maturity = 1,
  volatility = 0.3,
  dividend_yield = 0,
  payoff = "call",
  greek = c("fair_value", "delta", "rho", "vega", "theta", "gamma"),
  model = "black_scholes",
  lambda = 0.2,
  alpha = 0.3,
  jump_distribution = function(n) stats::rt(n, df = 3),
  steps = round(time_to_maturity * 252),
  paths = 10000,
  seed = 1,
  antithetic = FALSE
)
```

Arguments

- | | |
|----------------|---|
| initial_price | • initial price of the underlying asset, can also be a vector |
| exercise_price | • strike price of the option, can also be a vector |
| r | • risk-free interest rate |

<code>time_to_maturity</code>	<ul style="list-style-type: none"> time to maturity in years
<code>volatility</code>	<ul style="list-style-type: none"> volatility of the underlying asset
<code>dividend_yield</code>	<ul style="list-style-type: none"> dividend yield
<code>payoff</code>	<ul style="list-style-type: none"> the payoff function, either a string in ("call", "put", "digital_call", "digital_put"), or a function
<code>greek</code>	<ul style="list-style-type: none"> the Greek to be calculated
<code>model</code>	<ul style="list-style-type: none"> the model to be chosen in ("black_scholes", "jump_diffusion")
<code>lambda</code>	<ul style="list-style-type: none"> the lambda of the Poisson process in the jump-diffusion model
<code>alpha</code>	<ul style="list-style-type: none"> the alpha in the jump-diffusion model influences the jump size
<code>jump_distribution</code>	<ul style="list-style-type: none"> the distribution of the jumps, choose a function which generates random numbers with the desired distribution
<code>steps</code>	<ul style="list-style-type: none"> the number of integration steps
<code>paths</code>	<ul style="list-style-type: none"> the number of simulated paths
<code>seed</code>	<ul style="list-style-type: none"> the seed of the random number generator
<code>antithetic</code>	<ul style="list-style-type: none"> if TRUE, antithetic random numbers will be chosen to decrease variance

Value

Named vector containing the values of the Greeks specified in the parameter `greek`.

Examples

```
Malliavin_Asian_Greeks(initial_price = 110, exercise_price = 100,
r = 0.02, time_to_maturity = 4.5, dividend_yield = 0.015, volatility = 0.22,
greek = c("fair_value", "delta", "rho"), payoff = "put")
```

Index

Binomial_American_Greeks, [2](#)
BS_European_Greeks, [3](#)
BS_Geometric_Asian_Greeks, [4](#)
BS_Implied_Volatility, [5](#)
BS_Malliavin_Asian_Greeks, [6](#)

Greeks, [7](#)
Greeks_UI, [9](#)

Implied_Volatility, [9](#)

Malliavin_Asian_Greeks, [10](#)
Malliavin_European_Greeks, [12](#)
Malliavin_Geometric_Asian_Greeks, [13](#)