Package ‘greeks’

February 23, 2024

Title Sensitivities of Prices of Financial Options and Implied Volatilities

Version 1.3.4

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

Roxygen list(markdown = TRUE)

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

URL `https://github.com/ahudde/greeks`

BugReports `https://github.com/ahudde/greeks/issues`

NeedsCompilation yes

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

Maintainer Anselm Hudde <anselmhudde@gmx.de>

VignetteBuilder knitr

R topics documented:

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`.

See Also

Greeks_UI for an interactive visualization

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

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`.

See Also

Malliavin_European_Greeks for the Monte Carlo implementation

Greeks_UI for an interactive visualization

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

`initial_price`	• initial price of the underlying asset, can also be a vector
`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`	• the payoff function, either a string in ("call", "put")
`greek`	• 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`.

See Also

Malliavin_Geometric_Asian_Greeks for the Monte Carlo implementation which provides digital and custom payoff functions and also works for the jump diffusion model

Greeks_UI for an interactive visualization

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 put- and call options in the Black Scholes model via Halley's method.*

Description

Computes the implied volatility for European put- and call 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.

See Also

Implied_Volatility for American and Asian options, and for digital payoff functions

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", "rho", "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`.

See Also

Malliavin_Asian_Greeks for a greater set of Greeks and also in the jump diffusion model

Greeks_UI for an interactive visualization

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` *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*

Description

Greeks are derivatives of the option value with respect to underlying parameters. For instance, the Greek $\Delta = \frac{\partial \text{fair\_value}}{\partial \text{initial\_price}}$ (Delta) measures how the price of an option changes with a minor change in the underlying asset's price, while $\Gamma = \frac{\partial \text{fair\_value}}{\partial \text{initial\_price}}$ (Gamma) measures how Δ itself changes as the price of the underlying asset shifts. Greeks can be computed for different types of options: For

- **European Greeks** see also BS_European_Greeks and Malliavin_European_Greeks
- **American Greeks** see also Binomial_American_Greeks
- **Asian Greeks** see also BS_Malliavin_Asian_Greeks and Malliavin_Asian_Greeks
- **Geometric Asian Greeks** see also BS_Geometric_Asian_Greeks and Malliavin_Asian_Greeks

The Greeks are defined as the following partial derivatives of the option value:

- `Delta` = $\Delta = \frac{\partial \text{fair\_value}}{\partial \text{initial\_price}}$, the derivative with respect to the price of the underlying asset
- `Vega` = $\mathcal{V} = \frac{\partial \text{fair\_value}}{\partial \text{volatility}}$, the derivative with respect to the volatility
- `Theta` = $\Theta = -\frac{\partial \text{fair\_value}}{\partial \text{time\_to\_maturity}}$, the negative derivative with respect to the time until expiration of the option
- `rho` = $\rho = \frac{\partial \text{fair\_value}}{\partial r}$, the derivative with respect to the risk-free interest rate
- `Epsilon` = $\epsilon = \frac{\partial \text{fair\_value}}{\partial \text{time\_to\_maturity}}$, the derivative with respect to the dividend yield of the underlying asset
- `Lambda` = $\lambda = \Delta \times \frac{\text{initial\_price}}{\text{exercise\_price}}$
- `Gamma` = $\Gamma = \frac{\partial^2 \text{fair\_value}}{\partial \text{initial\_price}^2}$, the second derivative with respect to the price of the underlying asset
- `Vanna` = $\frac{\partial \Delta}{\partial \text{volatility}} = \frac{\partial^2 \text{fair\_value}}{\partial \text{intial\_price} \, \partial \text{volatility}}$, the derivative of Δ with respect to the volatility
- `Vomma` = $\frac{\partial^2 \text{fair\_value}}{\partial \text{volatility}^2}$, the second derivative with respect to the volatility
- `Veta` = $\frac{\partial \mathcal{V}}{\partial r} = \frac{\partial^2 \text{fair\_value}}{\partial \text{volatility} \, \partial \text{time\_to\_maturity}}$, the derivative of $\mathcal{V}$ with respect to the time until expiration of the option
- `Vera` = $\frac{\partial^2 \text{fair\_value}}{\partial \text{volatiliy} \, \partial \text{r}}$, the derivative of $\mathcal{V}$ with respect to the risk-free interest rate
- `Speed` = $\frac{\partial \Gamma}{\partial \text{initial\_price}} = \frac{\partial^3 \text{fair\_value}}{\partial \text{initial\_price}^3}$, the third derivative of the option value with respect to the price of the underlying asset
- `Zomma` = $\frac{\Gamma}{\text{volatility}} = \frac{\partial^3 \text{fair\_value}}{\partial \text{volatility}^3}$, the derivative of Gamma with respect to the volatility
- `Color` = $\frac{\partial \Gamma}{\partial \text{r}} = \frac{\partial^3 \text{fair\_value}}{\partial \text{initial\_price}^2 \, \partial \text{r}}$, the derivative of Gamma with respect to the risk-free interest rate
- `Ultima` = $\frac{\partial \text{Vomma}}{\partial \text{volatility}} = \frac{\partial^3 \text{fair\_value}}{\partial \text{volatility}^3}$, the third derivative with respect to the volatility

For the definitions of Greeks, and the different types of option see Hull (2022) or
en.wikipedia.org/wiki/Greeks_(finance)

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

`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

model
: • the model to be chosen in ("black_scholes", "jump_diffusion")

option_type
: in c("European", "American", "Asian", "Geometric Asian", "Digital", "Binomial) - the type of option to be considered

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")

antithetic
: • if TRUE, antithetic random numbers will be chosen to decrease variance

...
: • ... Other arguments passed on to methods

Value

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

References

Hull, J. C. (2022). Options, futures, and other derivatives (11th Edition). Pearson

en.wikipedia.org/wiki/Greeks_(finance)

See Also

BS_European_Greeks for option_type = "European"

Binomial_American_Greeks for option_type = "American"

BS_Geometric_Asian_Greeks for option_type = = "Geometric Asian" and model = "black_scholes"

BS_Malliavin_Asian_Greeks for option_type = = "Asian" and model = "black_scholes" and greek in c("fair_value", "delta", "rho", "vega")

Malliavin_Asian_Greeks for more general cases of Asian Greeks

Greeks_UI for an interactive visualization

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 — *Opens a shiny app to interactively visualize option prices and Greeks*

Description

Opens a shiny app to interactively visualize option prices and Greeks

Usage

```
Greeks_UI()
```

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

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.

See Also

BS_Implied_Volatility for the special case option_type = "European" and payoff in c("call", "put")

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

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", "digital_call", "digital_put"), or a function
greek	• the Greek to be calculated
model	• the model to be chosen in ("black_scholes", "jump_diffusion")
lambda	• the lambda of the Poisson process in the jump-diffusion model
alpha	• the alpha in the jump-diffusion model influences the jump size
jump_distribution	• the distribution of the jumps, choose a function which generates random numbers with the desired distribution
steps	• the number of integration steps
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.

See Also

BS_Malliavin_Asian_Greeks for a faster computation, but only in the Black Scholes model and with a smaller set of Greeks

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`

See Also

BS_European_Greeks for the exact and fast implementation for call-, put- and digital payoff functions

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 a geometric Asian option with the Malliavin Monte Carlo Method in the Black Scholes- or Jump diffusion model

---

Description

Computes the Greeks of a geometric Asian option with the Malliavin Monte Carlo Method in the Black Scholes- or Jump diffusion 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
`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", "digital_call", "digital_put"), or a function
greek	• the Greek to be calculated
model	• the model to be chosen in ("black_scholes", "jump_diffusion")
lambda	• the lambda of the Poisson process in the jump-diffusion model
alpha	• the alpha in the jump-diffusion model influences the jump size
jump_distribution	• the distribution of the jumps, choose a function which generates random numbers with the desired distribution
steps	• the number of integration steps
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`.

See Also

BS_Geometric_Asian_Greeks for exact and fast computation in the Black Scholes model and for put- and call payoff functions

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