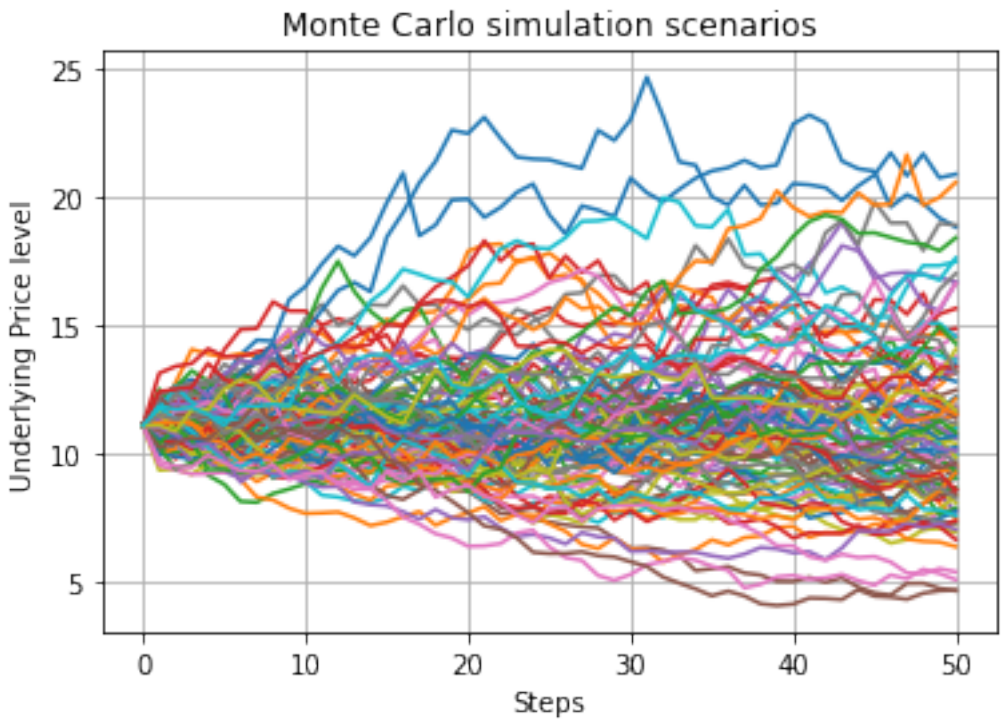
Option Trading Strategies

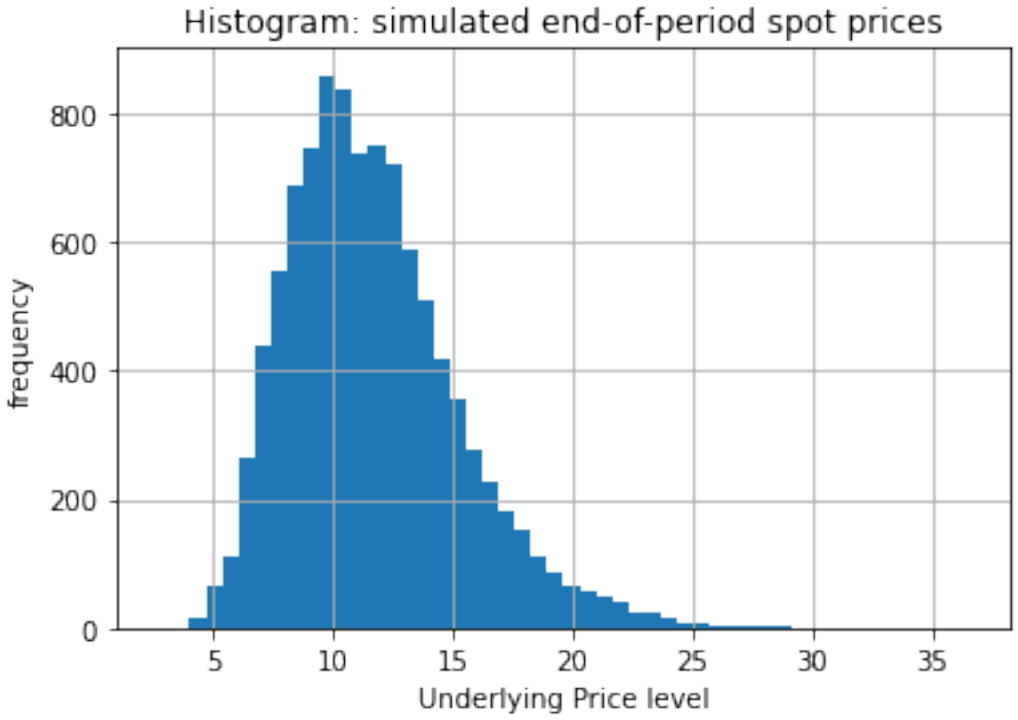
February 7, 2021

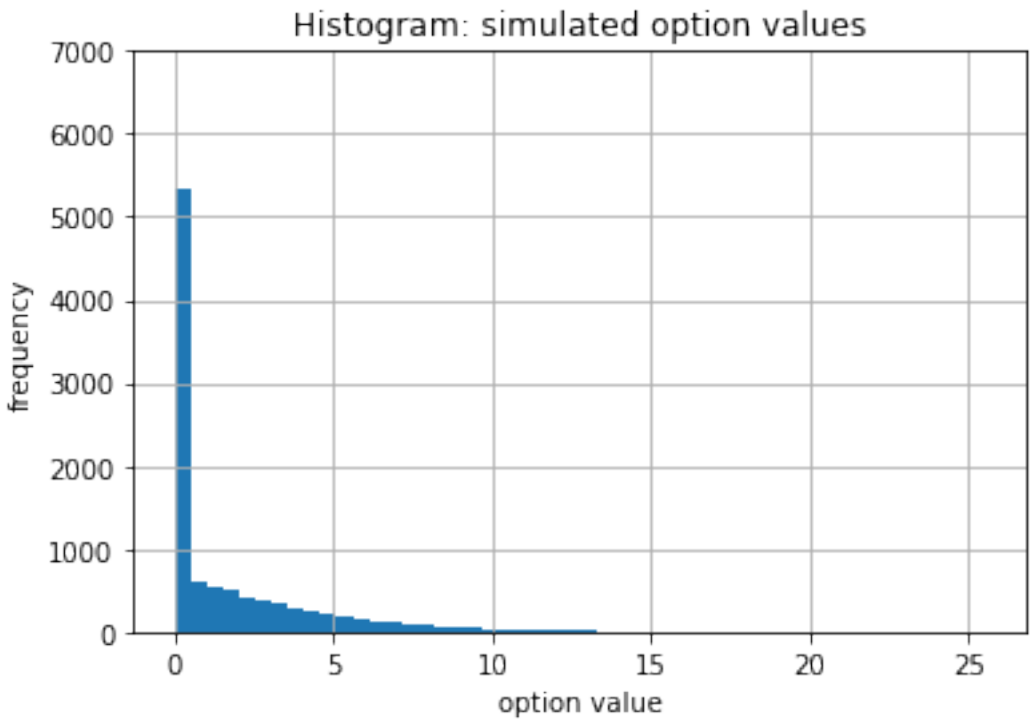
[1]:
```
# Implementation of Option strategies
# ------------------------------------------------------------------------------
# Program that constructs Option trading strategies using plain vanilla options
# Object Oriented Programming principles have been applied like Inheritance and
# Polymorphism.
# ------------------------------------------------------------------------------
# As an example "Long Butterfly" option strategy is implemented and the trade
# is the following:
# Buy one call at A,
# Sell two calls at higher strike B
# Buy one call at an even higher strike C.
# ------------------------------------------------------------------------------
# Market expectation: Direction neutral/volatility bearish. In this case,
# the holder expects the underlying to remain around strike B, or it is felt
# that there will be a fall in implied volatility. Position is less risky than
# selling straddles or strangles as there is a limited downside exposure.
# ------------------------------------------------------------------------------
# Profit & loss characteristics at expiry:
#
# Profit: Maximum profit limited to the difference in strikes between A and B
# minus the net cost of establishing the position. Maximised at mid strike B
# (assuming A-B and B-C are equal).
#
# Loss: Maximum loss limited to the net cost of the position for either a rise
# or a fall in the underlying.
#
# Break-even: Reached when the underlying is higher than A or lower than C by
# the cost of establishing the position.
#
# DEUTSCHE LUFTHANSA ORD with ISIN: DE0008232125 serves as the Underlying in
# this example.
# ------------------------------------------------------------------------------

import numpy as np
from Derivatives.Portfolio import Portfolio
from Derivatives.Options.Option import Option
from Derivatives.Options.PlainVanillaOption import PlainVanillaOption
```

[2]:
```
# Step 1 - Set up the options with the following parameters
# ---------------------------------------------------------------------------
# Spot price = 11.10,
# Time to maturity 1 year = 1
# Risk-free interest rate = 0.05
# Implied volatility = 0.30
# Multiplier = 100 for LHA options. Contract size 100 for long option, 200 for
# the short option
# Strike prices = Call A: 11, Call B = 9, Call C = 13
# ---------------------------------------------------------------------------
option_1 = PlainVanillaOption(11.10, 11, 1, 0.05, 0.30,'call','Short', 200, 100)
option_2 = PlainVanillaOption(11.10, 9, 1, 0.05, 0.30, 'call', 'Long', 100, 100)
option_3 = PlainVanillaOption(11.10, 13, 1, 0.05, 0.30, 'call','Long', 100, 100)
```

[3]:
```
#Step 2 - Price individually the options and calculate their risk measures.
# The options will then be added in the portfolio
# ---------------------------------------------------------------------------
# Option 1: Short call @11, 200 contracts of 100 options each

# Pricing using the closed-end formula of B&S. Uncomment for usage.
#option_1.calculate_option_price_BS_formula()

# In this example the MC method has been used for the pricing of the Option 1
option_1.calculate_option_price_MC_BS()
# Calculate Option 1 greeks: Delta, Gamma, Vega
option_1.calculate_option_greeks()
 # Get all the information for Option 1. Uncomment to use
option_1.get_option_properties()
```


Monte Carlo simulation scenarios
Underlying Price level
Steps
25
20
15
10
5
0
10
20
30
40
50

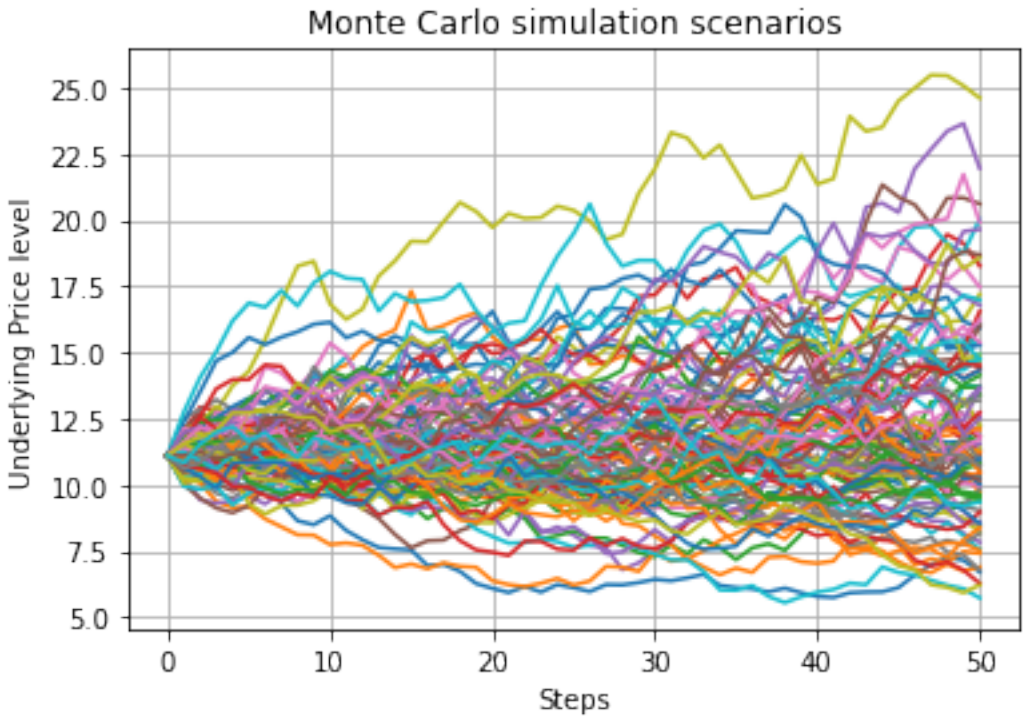

Histogram: simulated end-of-period spot prices
frequency
Underlying Price level
800
600
400
200
0
5
10
15
20
25
30
35

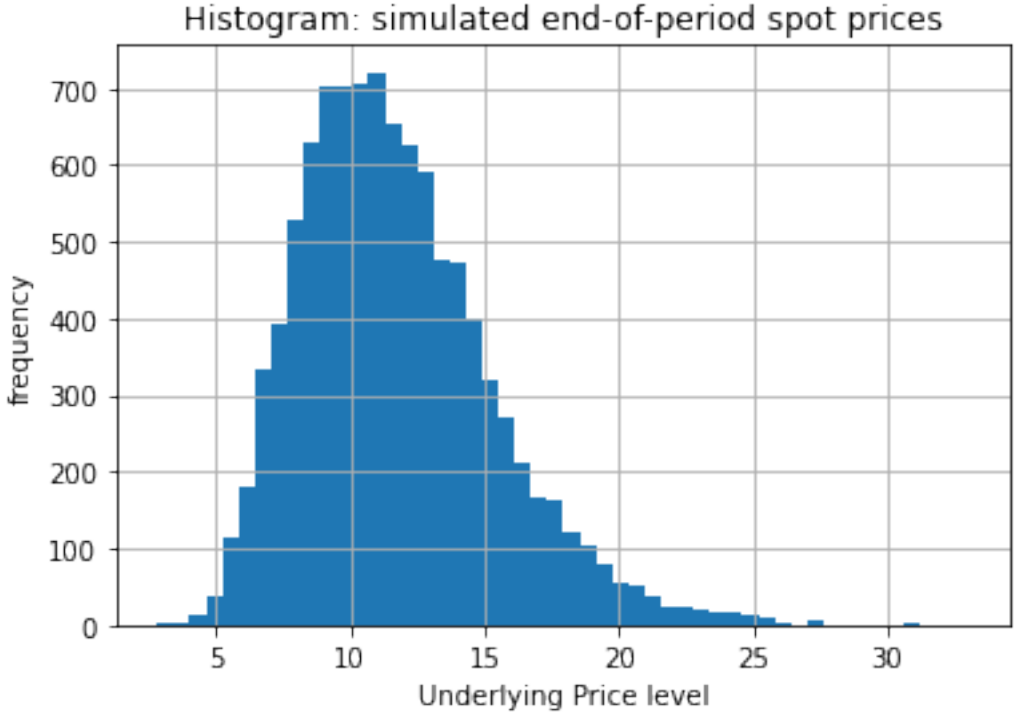


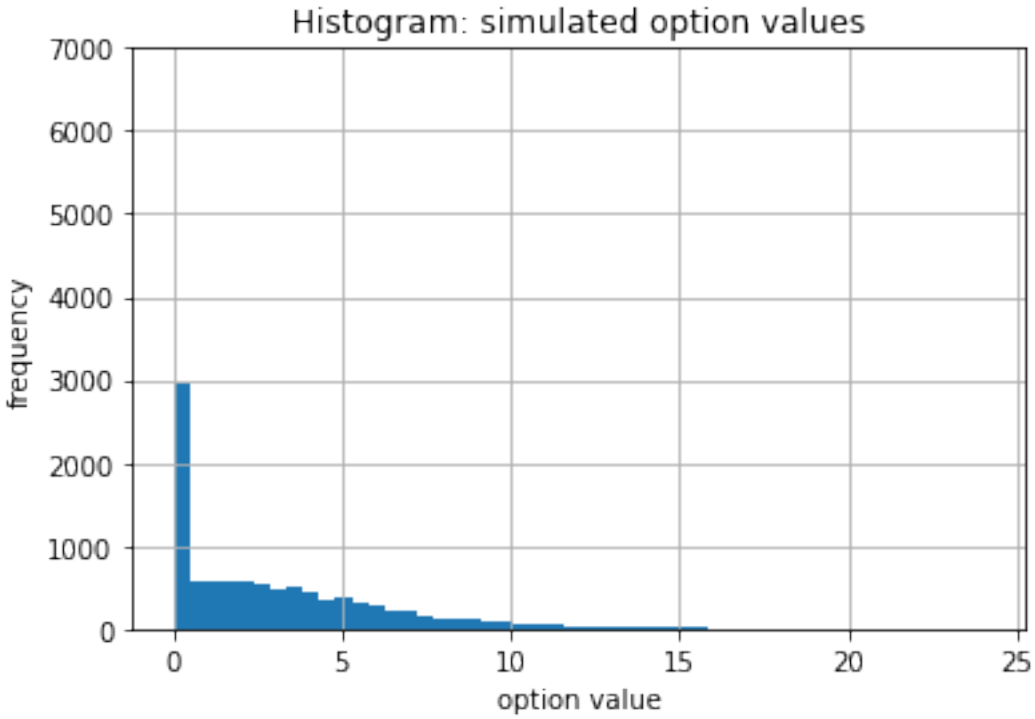

```
Properties of the call @ Strike:  11
-----------------------------------
S0 : 11.1
K : 11
T : 1
r : 0.05
position_type : Short
contract_size : 200
multiplier : 100
s : 0.3
option_type : call
price : 1.64262395271364
delta : 0.6356415273675872
gamma : 0.1128092062456668
vega : 4.169766690458582
```

[4]:
```
# ------------------------------------------------------------------------------
# Option 2: Long call @9, 100 contracts of 100 options each

# In this example the MC method has been used for the pricing of the Option 2
option_2.calculate_option_price_MC_BS()
# Calculate Option 2 greeks: Delta, Gamma, Vega
option_2.calculate_option_greeks()
# Get all the information for Option 1. Uncomment to use
```

```
option_2.get_option_properties()
```



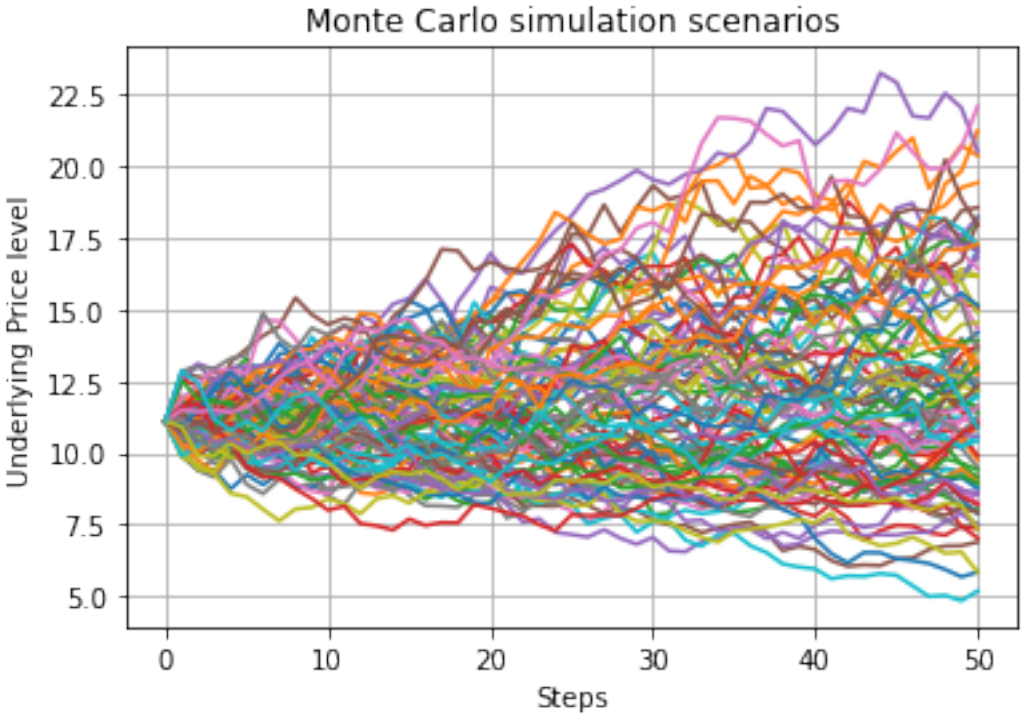


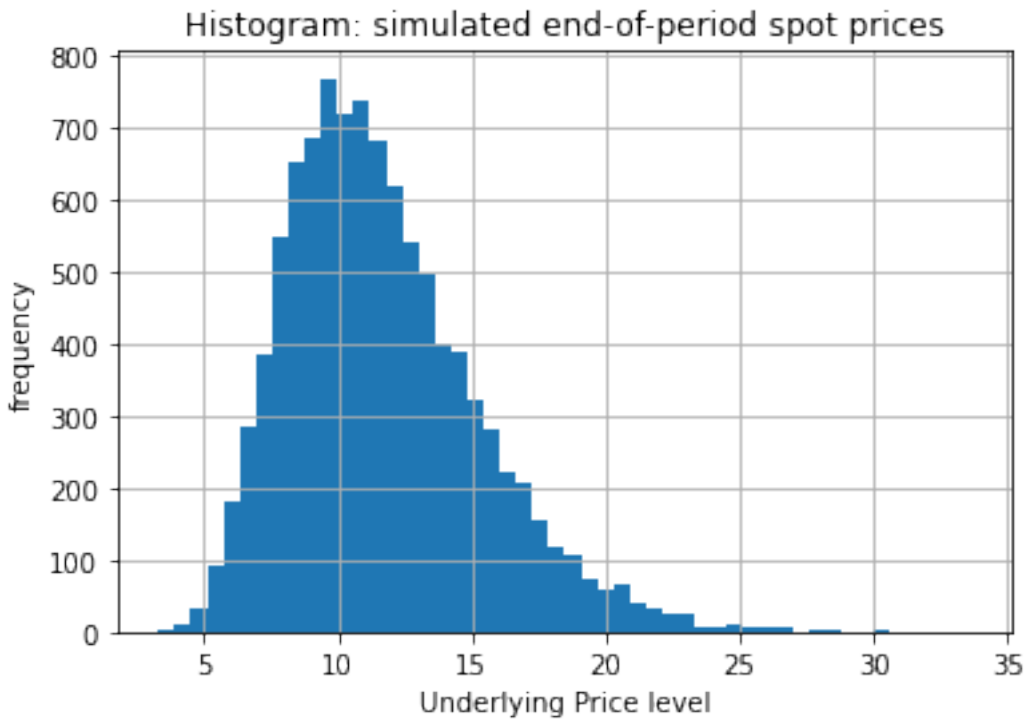


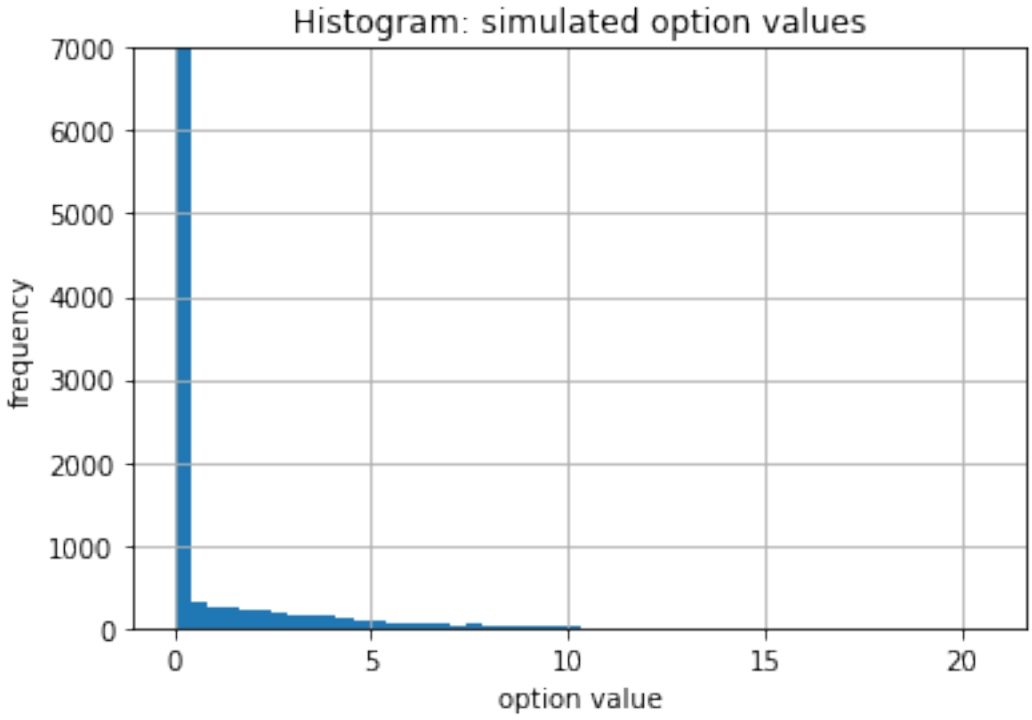

```
Properties of the call @ Strike:  9
-----------------------------------
S0 : 11.1
K : 9
T : 1
r : 0.05
position_type : Long
contract_size : 100
multiplier : 100
s : 0.3
option_type : call
price : 2.9001085634637365
delta : 0.8451222264539587
gamma : 0.07152060155072537
vega : 2.643615995119461
```

[5]:
```
# ------------------------------------------------------------------------
# Option 3: Long call @13, 100 contracts of 100 options each

# In this example the MC method has been used for the pricing of the Option 3
option_3.calculate_option_price_MC_BS()
# Calculate Option 3 greeks: Delta, Gamma, Vega
```

```
option_3.calculate_option_greeks()
# Get all the information for Option 1. Uncomment to use
option_3.get_option_properties()
```




Histogram: simulated end-of-period spot prices
frequency
Underlying Price level
0
100
200
300
400
500
600
700
800
5
10
15
20
25
30
35


Histogram: simulated option values
frequency
option value
0
1000
2000
3000
4000
5000
6000
7000
0
5
10
15
20

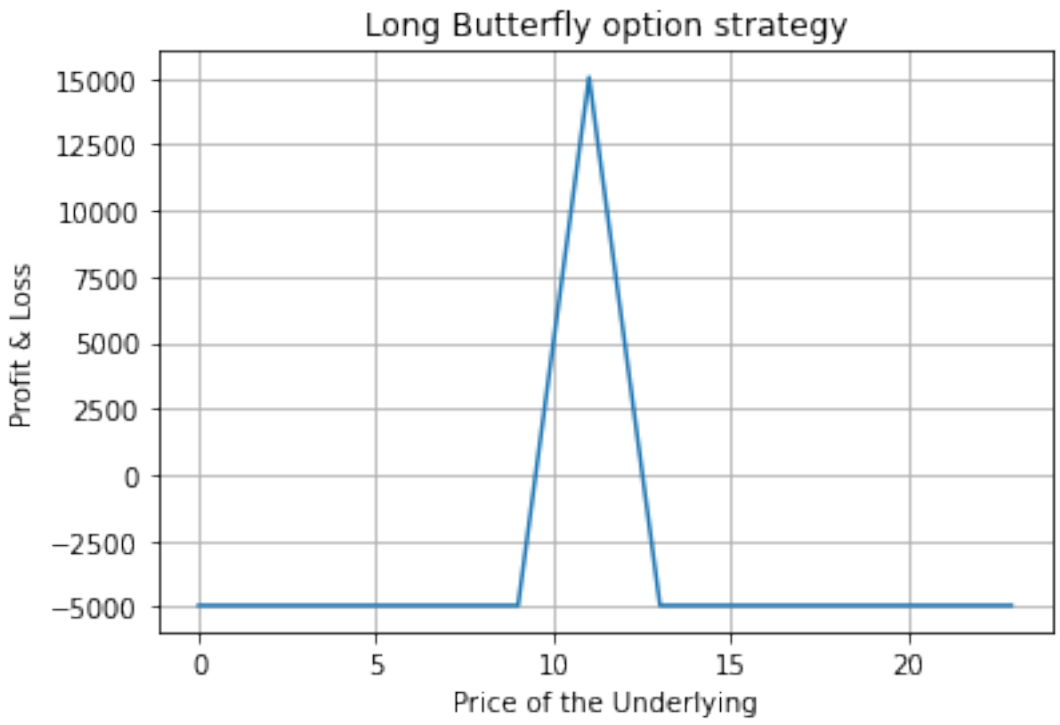
```
Properties of the call @ Strike:  13
-----------------------------------
S0 : 11.1
K : 13
T : 1
r : 0.05
position_type : Long
contract_size : 100
multiplier : 100
s : 0.3
option_type : call
price : 0.8804983868230675
delta : 0.4168283092136471
gamma : 0.11718940444357637
vega : 4.331671956447913
```

[6]:
```
#Step 3 - Create the empty portfolio and add the options in it
portfolio = Portfolio('Long Butterfly')
portfolio.add_product(option_1)
portfolio.add_product(option_2)
portfolio.add_product(option_3)
```

[7]:
```
portfolio.calculate_premium_strategy()
```

```
The premium for the strategy stands at:  -4953.59  Euros
```

[8]:
```
portfolio.calculate_payoff_strategy()
```



[9]:
```
# ------------------------------------------------------------------------
# As a conclusion we total premium paid for entering in this strategy is around
# 4,950 Euros and this amount is the total risk of the strategy.
# The maximum gain from the strategy could be approximately 15,050 Euros.
```

[ ]: