**Introduction to Options and Option pricing using "quantsbin" library**

**PART 1**

In this part of te tutorial we will explore below topics:

1. Installing Quantsbin

2. Defining Option objects

3. Calculating and plotting payoff

4. Creating engine object to pass market data to option object

5. Selecting model for valuation

6. Calculating and plotting option premium and greeks

For full set of features, please look at Quantsbin [Documentation](http://www.quantsbin.com/documentation/) page

**Installation and Import**

Let's start with installing Python open source library Quantsbin: Quantsbin library could be installed using either of the below method -

1. PIP installation: Simply type <pip install quantsbin> in your terminal

2. Download package from [Github](https://github.com/quantsbin/Quantsbin)

Once **Quantsbin** is installed, import the Derivative pricing module and assign qbdp as alias to it.

In [17]:

**import** **quantsbin.derivativepricing** **as** **qbdp**

**Defining Option Objects**

In current release **Quantsbin** provide options to define under four asset classes. Options on:

1. Equity: qbdp.EqOption

2. Futures: qbdp.FutOption

3. Currencies: qbdp.FXOption

4. Commodities: qbdp.ComOption

Let's check doc string to get more information about Option objects.

In [18]:

print(qbdp.EqOption.\_\_doc\_\_)

Defines object for vanilla options on equity with both European and American expiry type.

Args required:

option\_type: 'Call' or 'Put' (default value is set to 'Call')

expiry\_type: 'European' or 'American' (default is set to 'European')

strike: (Float in same unit as underlying price) e.g. 110.0

expiry\_date: (Date in string format "YYYYMMDD") e.g. 10 Dec 2018 as "20181210"

derivative\_type: Default value as "Vanilla Option".

We can observe above that therevare five argumnents which are required for deifning an option.

Let's deifne our own Equity call option with European expiry.

In [19]:

equity\_option1 = qbdp.EqOption(option\_type='Call', strike=50, expiry\_date='20190621')

**Congrates!!!** you have defined you first option. Let's explore more into this option.

We will start with calculating payoff given Spot (current underlying price) is 55.

In [20]:

print(equity\_option1.payoff(55))

5

**Plotting Payoff**

Let's plot the payoff profile for this option.

qbdp.Plotting takes first argument as option object, second argument as parameter to be plotted and this argument as range for which plot is required.

**Payoff Plotting**

In [21]:

eq1\_payoff = qbdp.Plotting(equity\_option1,'payoff',x\_axis\_range=[25,75]).line\_plot()

eq1\_payoff.show()



**Option Pricing and Greeks Calculation**

To price option we need two data points: Pricing Model and Market data. We pass these two parameters to our option using engine method which returns engine object.

To start with we are using BlackScholesMerton as our pricing model and updating all other market parameters in engine.

In [22]:

eq1\_engine = equity\_option1.engine(model='BSM',pricing\_date='20180620',spot0=55, rf\_rate=0.05, volatility=0.25)

Calculating **Option Premium**

In [23]:

print(eq1\_engine.valuation())

9.66280048534

Calculating **Option Greeks**

In [24]:

print(eq1\_engine.risk\_parameters())

{'delta': 0.75995706023859544, 'gamma': 0.022580254309896741, 'theta': -0.01025008645649733, 'vega': 17.123101752878206, 'rho': 32.222878479364951, 'phi': -41.912152390692953}

We can also check what all models are available for valuation by using below command

In [25]:

print(equity\_option1.list\_models())

BSM, MC\_GBM, Binomial

Once we have engine object ready, we can start by plotting pnl profile. PnL is simply payoff minus current option premium

Note: We need to pass pricing object for pnl plot, unlike payoff plot where we pass option object

**PnL Plotting**

In [26]:

eq1\_pnl = qbdp.Plotting(eq1\_engine, 'pnl', x\_axis\_range=[25,75]).line\_plot()

eq1\_pnl.show()



**Plotting Premium and Payoff together**

In [27]:

eq1\_payoff = qbdp.Plotting(equity\_option1,'payoff',x\_axis\_range=[25,75]).line\_plot()

eq1\_pnl = qbdp.Plotting(eq1\_engine, 'valuation', x\_axis\_range=[25,75]).line\_plot()

eq1\_pnl.show()



**Plotting Greeks**

**Delta**

In [28]:

eq1\_delta = qbdp.Plotting(eq1\_engine, 'delta', x\_axis\_range=[25,75]).line\_plot()

eq1\_delta.show()



**Gamma**

In [29]:

eq1\_delta = qbdp.Plotting(eq1\_engine, 'gamma', x\_axis\_range=[25,75]).line\_plot()

eq1\_delta.show()



**Theta**

In [30]:

eq1\_delta = qbdp.Plotting(eq1\_engine, 'theta', x\_axis\_range=[25,75]).line\_plot()

eq1\_delta.show()



**Vega**

In [31]:

eq1\_delta = qbdp.Plotting(eq1\_engine, 'vega', x\_axis\_range=[25,75]).line\_plot()

eq1\_delta.show()



**Rho**

In [32]:

eq1\_delta = qbdp.Plotting(eq1\_engine, 'rho', x\_axis\_range=[25,75]).line\_plot()

eq1\_delta.show()



**In next tutorial we will further deep dive into valuation of options on different asset classes and available market parameters to tweek**