OptionsL1.BlackScholes

March 25, 2016

0.1 Black-Scholes-Merton model

0.1.1 Предположения модели

В моделе Блэка-Шульца предполагается, что на рынке есть безрисковый актив (Bond) и как минимум один актив с риском (Stock). (B, S)-рынок.

Предположения о рынке:

- **No arbitrage** (не существует стратегии, которая бы давала прибыль без риска принести убытки).
- Можно занимать поцию с **любым номиналом** по все доступным активам. Включаю отрицательные (long / short selling, borrow / lend) и дробные номиналы.
- **Frictionless market**. Принебрегаем стоимостью транзаций и Bid/Ask spread.

Активы (B, S):

- Risk-free rate: $\mathbf{r} = \mathbf{const}$.
- Движение рыночной цены S - **геометрическое броуновское движение** с постоянными параметрами drift и volatility.
- В классическом варианте по underlying активу нет начисления дивидентов. Но модель легко обобщается на случай актива с дивидентами. Это необходимо для, например, для FX options.

Обозначения

- S - текущая цена базового актива
- t - текущее время
- T - maturity time
- K - strike price
- r - безрисковая процентная ставка (risk-free interest rate)
- q - процентная ставка выплат по базовому активу
- σ - volatility - параметр броуновкого движения выплат по базовому активу
- C - цена Европейского call-опциона

0.1.2 Общий ход рассуждений

(для случая $q = 0$) для Европейского call-опциона

Рассматривается портфель

$$\Pi = -C + \frac{\partial C}{\partial S} S$$

Можно показать, что в предположении что S это GBM, $d\Pi$ не зависит от случайной переменной, т.е. портфель - безрисковый актив. Этот прием называется delta-hedge.

Уровнение для цены опциона:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS\frac{\partial C}{\partial S} - rC = 0$$

PDE, not SDE (!)

$$C(S, K, t, T, r, \sigma) = N(d_+)S - N(d_-)Ke^{-r(\tau)}$$

Где $\tau = T - t$.

$N(z) = \Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\frac{x^2}{2}}dx$ это функция распределения для стандартного нормального (гаусовского) распределения.

Плотность распределения: $N'(z) = \phi(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$.

Вывод и решение уравнения можно посмотреть литературе.

0.1.3 Формула ($q \neq 0$)

Формула Блэка-Шульца для Европейского call-опциона:

$$C(S, K, \tau, r, q, \sigma) = N(d_+)Se^{-q\tau} - N(d_-)Ke^{-r\tau}$$

для put-опциона:

$$P(S, K, \tau, r, q, \sigma) = Ke^{-r\tau}N(-d_-) - S_0e^{-q\tau}N(-d_+)$$

где:

$$d_+ = \frac{\log\frac{S}{K} + (r - q + \frac{\sigma^2}{2}\tau)}{\sigma\sqrt{\tau}} = d_1$$

и

$$d_- = d_+ - \sigma\sqrt{\tau} = \frac{\log\frac{S}{K} + (r - q - \frac{\sigma^2}{2}\tau)}{\sigma\sqrt{\tau}} = d_2$$

In [1]:
```
import numpy as np
import scipy as sp
from scipy import stats
import math as m

import matplotlib.pyplot as plt
%matplotlib inline

#import pandas as pd
```

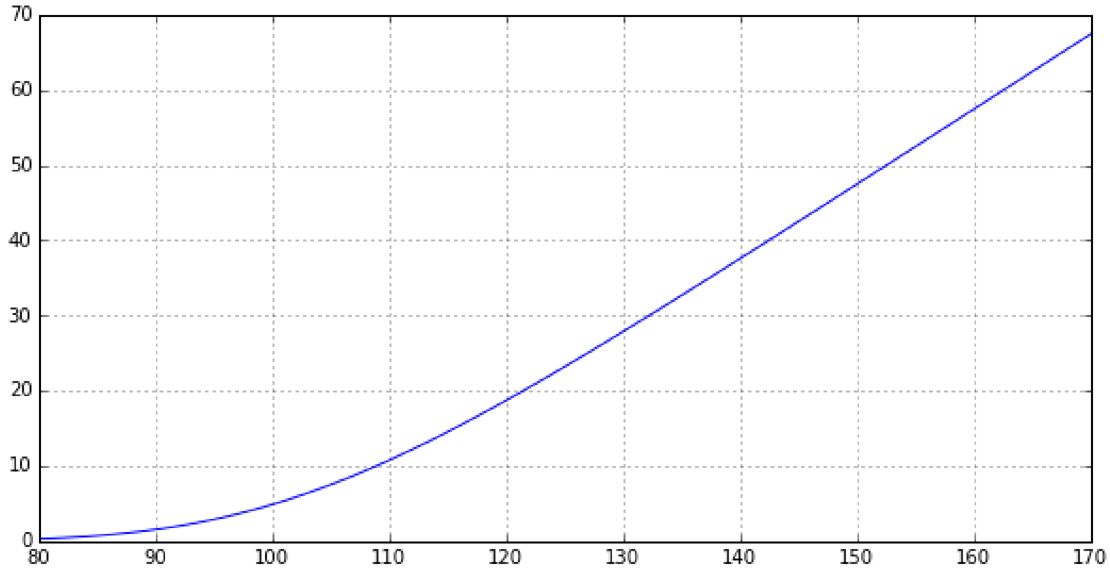
In [2]:
```
def bsm_call_price( S, K, T, r, sigma ):
  d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * m.sqrt(T))
  d2 = (np.log(S / K) + (r - 0.5 * sigma ** 2) * T) / (sigma * m.sqrt(T))
  return (S * stats.norm.cdf(d1, 0.0, 1.0) - K * m.exp(-r * T) * stats.norm.cdf(d2, 0.0, 1.0))

def call_payoff(K, S):
    return np.maximum(S-K, 0)
```

In [4]:
```
K = 110
r = 0.07
sigma = 0.15
x = np.linspace(K-30, K+60, num=100)
#
```

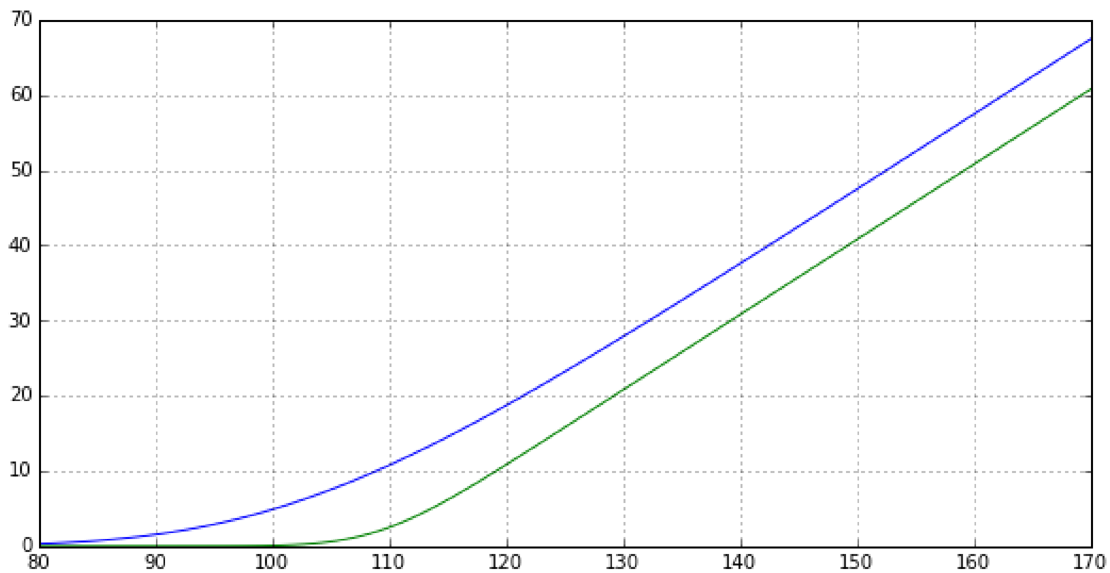
```
f, ax = plt.subplots(figsize=(10,5))
ax.grid(True)
ax.plot(x, bsm_call_price( x, K, 1, r, sigma ))
```

Out[4]: [<matplotlib.lines.Line2D at 0x7f8e3f6c2cc0>]



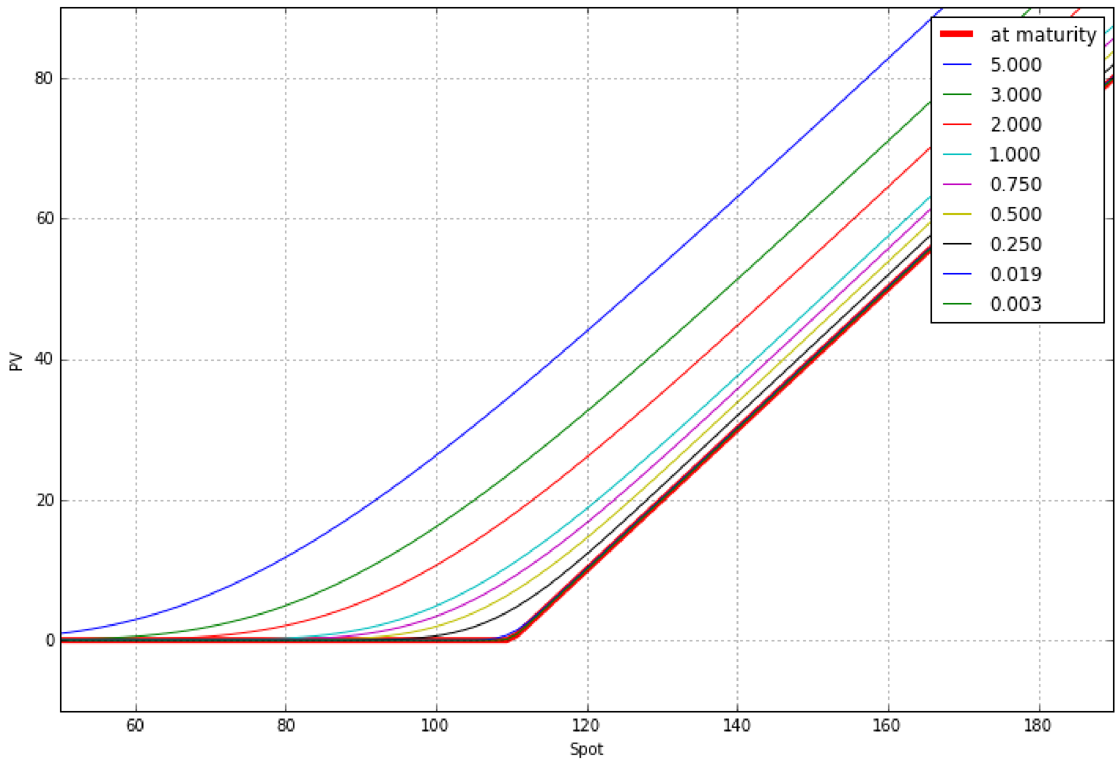
In [5]:
```
ax.plot(x, bsm_call_price( x, K, 0.1, r, sigma ))
f
```

Out[5]:



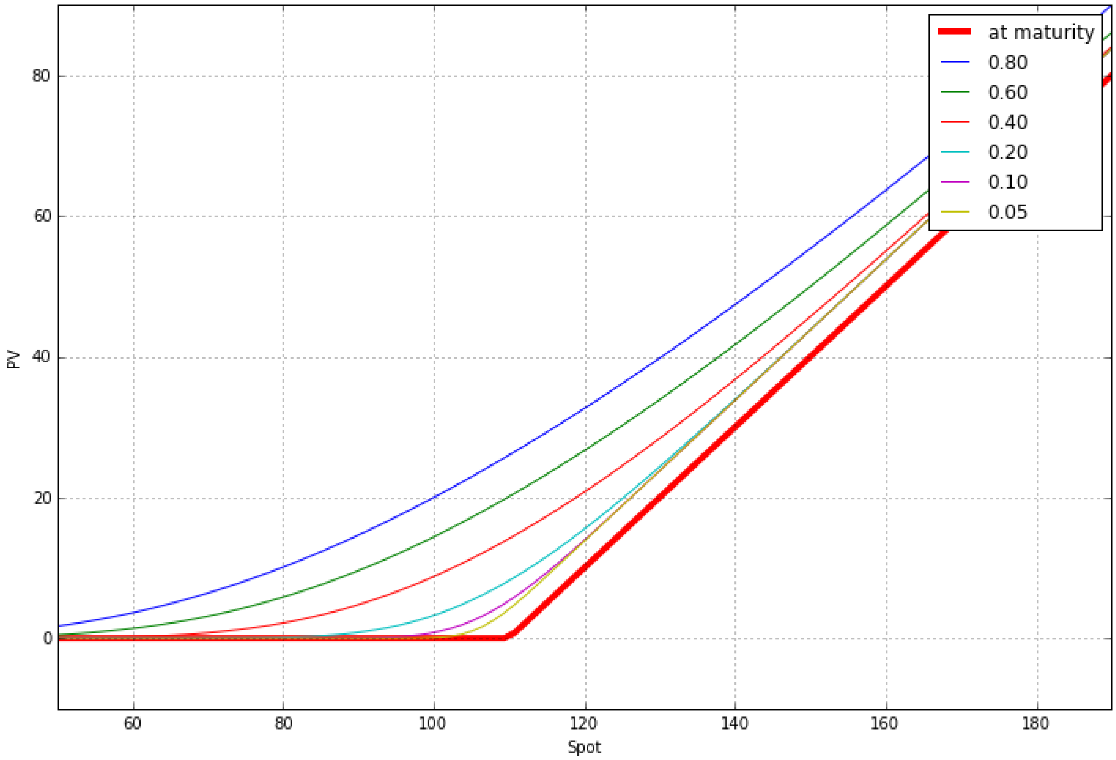
```
In [6]: Ts = [5, 3, 2, 1, 0.75, 0.5, 0.25, 1./52., 1./365.]

        x = np.linspace(50, K+80, num=100)
        f, ax = plt.subplots(figsize=(12,8))
        ax.grid(True)
        plt.plot(x, call_payoff(K, x), 'r', linewidth=4)
        for T in Ts:
            ax.plot(x, bsm_call_price( x, K, T, r, sigma ))
        plt.legend(['at maturity'] + ['{:.3f}'.format(t) for t in Ts])
        axes = plt.gca()
        axes.set_xlim([min(x),max(x)]); axes.set_ylim([-10,90]);
        axes.set_xlabel('Spot'); axes.set_ylabel('PV');
```




```
In [7]: T = 0.5
        sigmas = [0.8, 0.6, 0.4, 0.2, 0.1, 0.05]

        x = np.linspace(50, K+80, num=100)
        f, ax = plt.subplots(figsize=(12,8))
        ax.grid(True)
        plt.plot(x, call_payoff(K, x), 'r', linewidth=4)
        for sigma in sigmas:
            ax.plot(x, bsm_call_price( x, K, T, r, sigma ))
        plt.legend(['at maturity'] + ['{:.2f}'.format(s) for s in sigmas])
        axes = plt.gca()
        axes.set_xlim([min(x),max(x)]); axes.set_ylim([-10,90]);
        axes.set_xlabel('Spot'); axes.set_ylabel('PV');
```


at maturity
0.80
0.60
0.40
0.20
0.10
0.05
PV
Spot
80
60
40
20
0
60
80
100
120
140
160
180

```
In [ ]:
```