# JUPYTER 활용법

과제:

www.github.com 에 계정을 만들고 프로젝트를 생성한 후 anaconda 로 설치한 Jupyter를 구동하여 URP-BS-formula 내용을 정리해 봅니다.

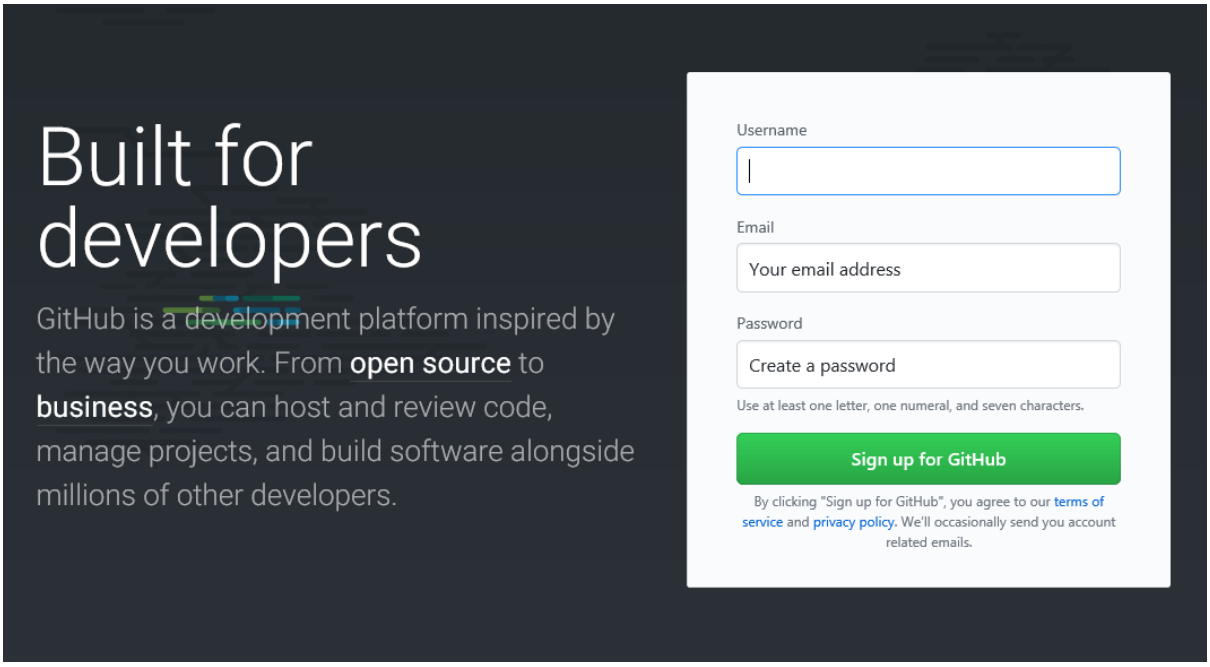
편한 username 으로 github 에 회원가입하고 URP-FinanceProject 라는 repository 를 개설합니다.

여기에 https://github.com/aorc-group2/URP-FinanceProject 에 주어진

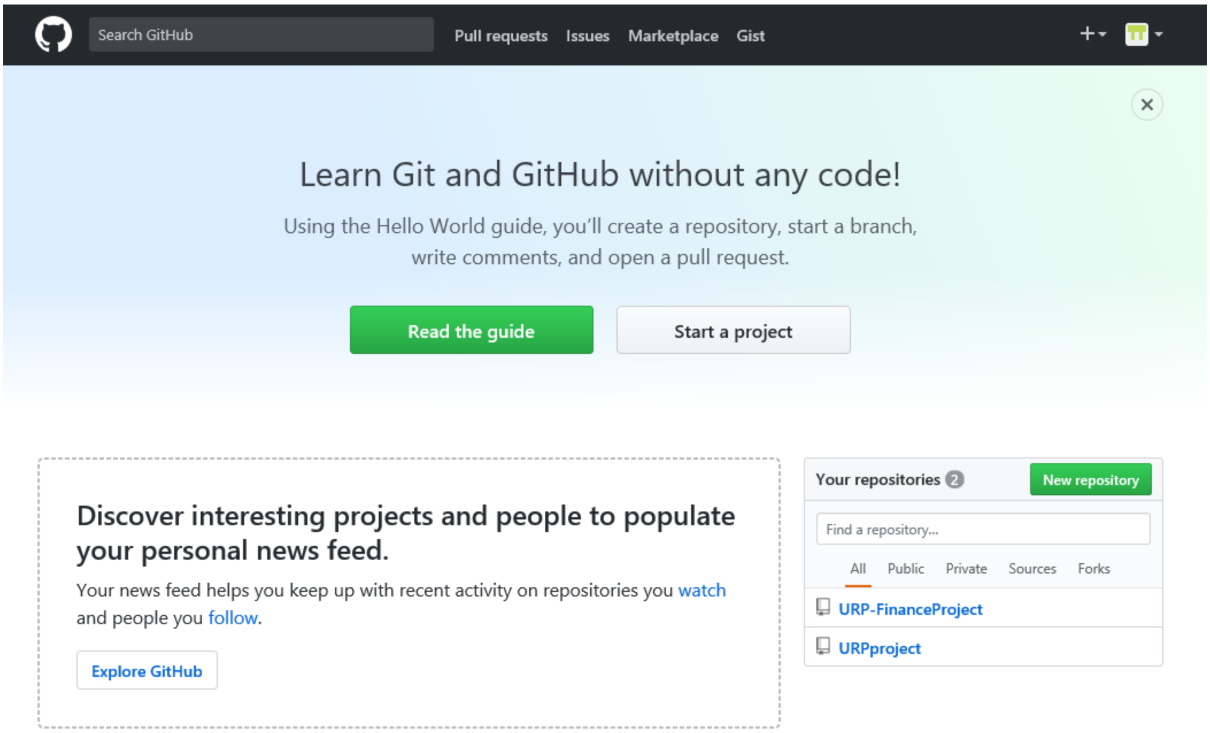
BSformula_Haug.py 와 URP-BS-example.txt 를 Jupyter 를 통하여 URP-BS-example.ipynb 파일로 저장하고 BSformula_Haug.py 와 URP-BS-example.txt 그리고 URP-BS-example.ipynb 파일을 업로드 해서 위 https://github.com/aorc-group2/URP-FinanceProject 와 같은 프로젝트를 각자의 유저 계정에 만들어 봅니다.

프로젝트 완성 후 github 주소를 내게 이메일로 답장 주세요.

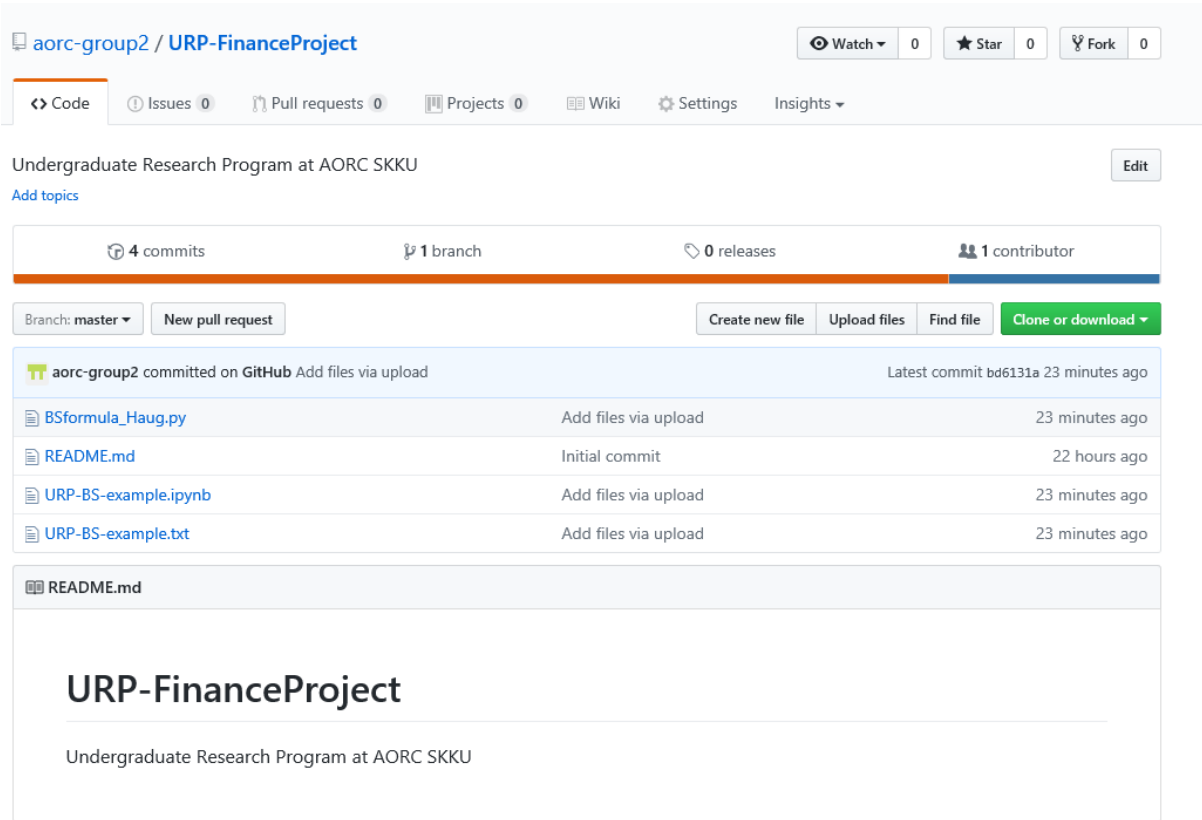
1. Github 에서 계정을 만듭니다. Username 은 각자 편한 이름을 고르고 이메일을 확인하면 sign up 됩니다.



제 계정에는 URPproject 와 URP-FinanceProject 가 있습니다.




아래와 같은 repository를 각자의 github.com/"username"/URP-FinanceProject 로 만드는게 과제입니다.

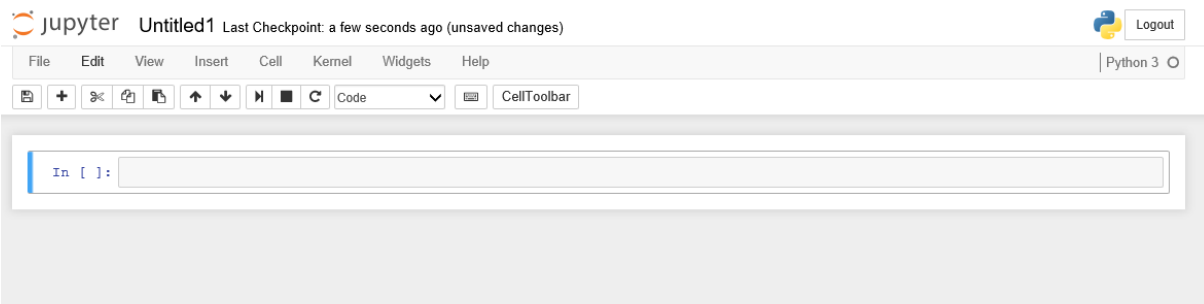


이제 URP-BS-example.ipynb (ipython notebook format) 을 만들어 봅시다.

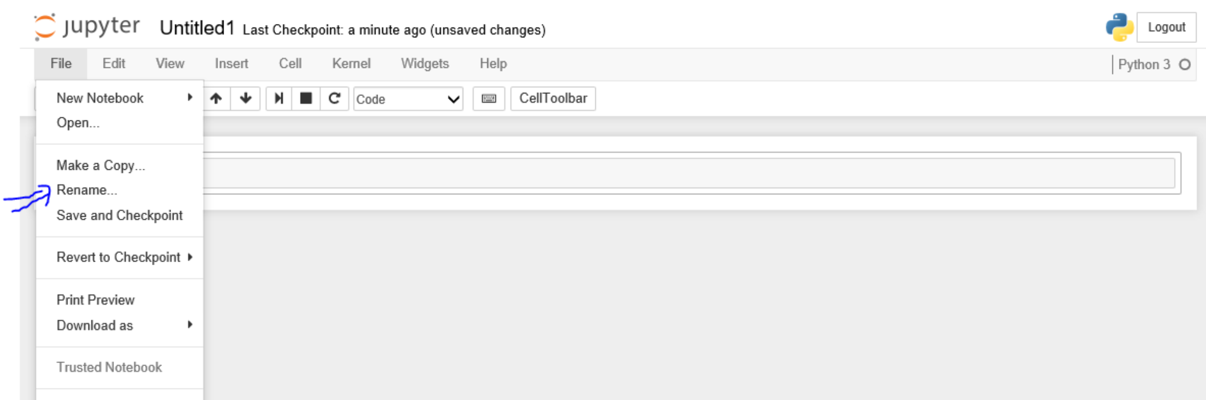
Anaconda를 통해서 Jupyter 를 실행합니다.

New menu 에서 python3 notebook 을 새로 만듭니다.

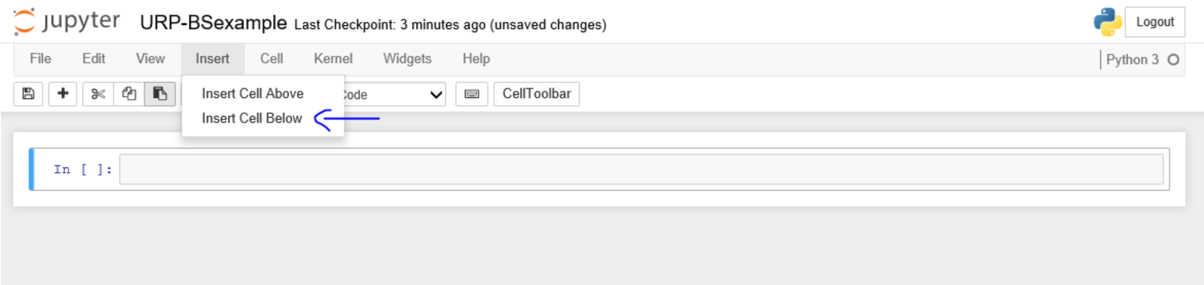

아래처럼 새 노트북이 untitled 로 만들어집니다.




원하는 이름으로 바꿉시다. 우리는 URP-BS-example

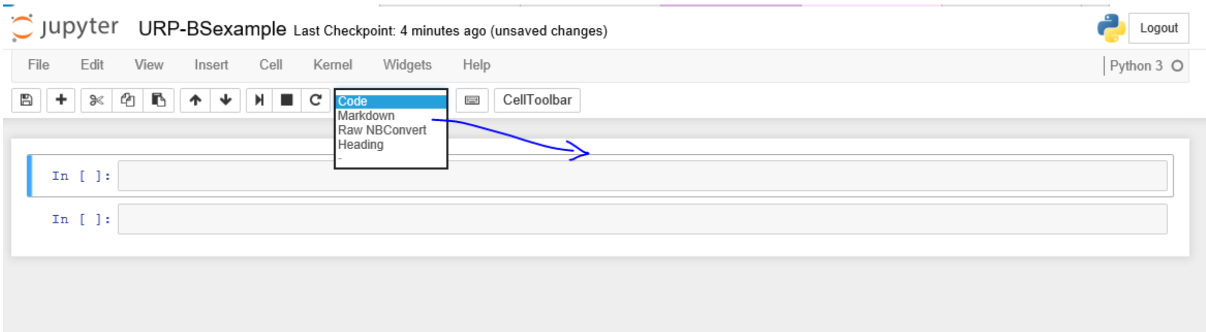


내용 한 줄과 실행내역이 필요해서 한 줄을 추가합니다. (간단히 return 누르면 한 줄 생깁니다 사실...)

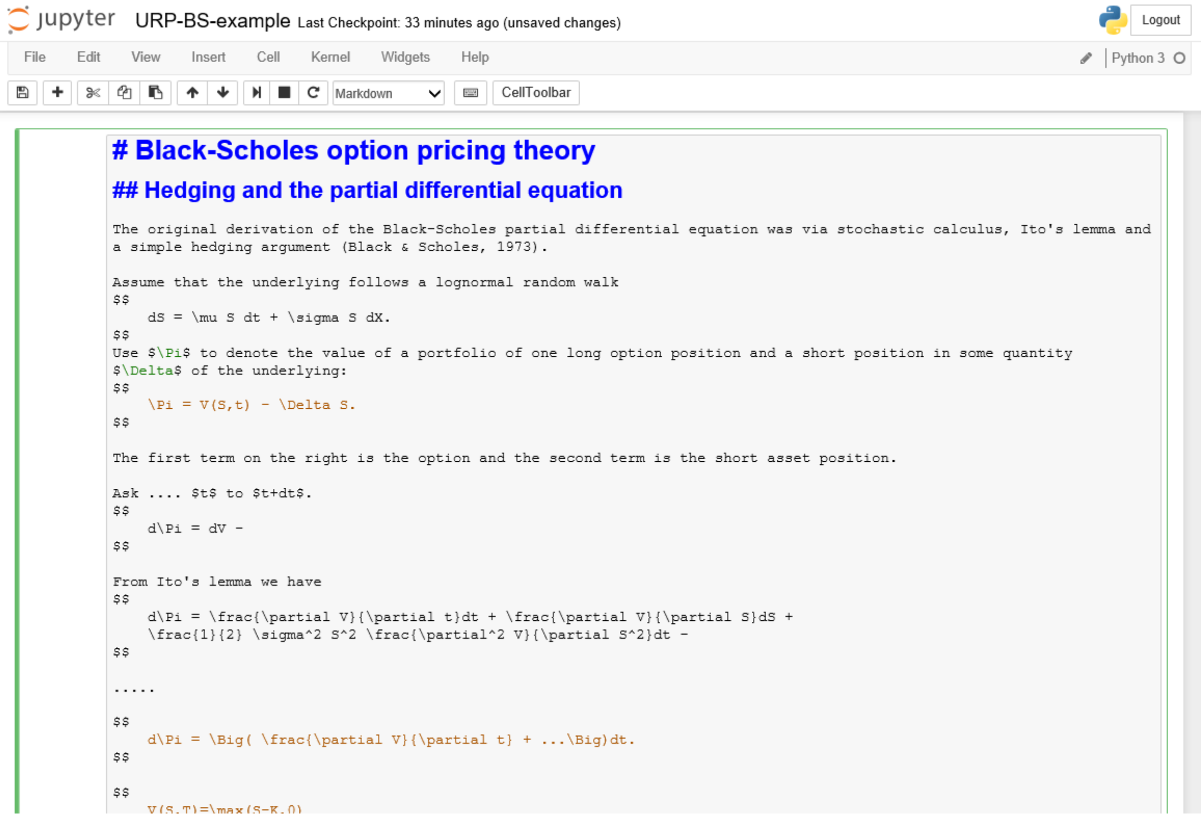



Jupyter cell 은 크게code 와 markdown 으로 설정할 수 있습니다. 우리는 code 에서는 python 프로그램을 실행하고, markdown 에서는 latex 을 활용해서 내용을 정리합니다.

윗 줄의 형식을 markdown  으로 변경합니다.

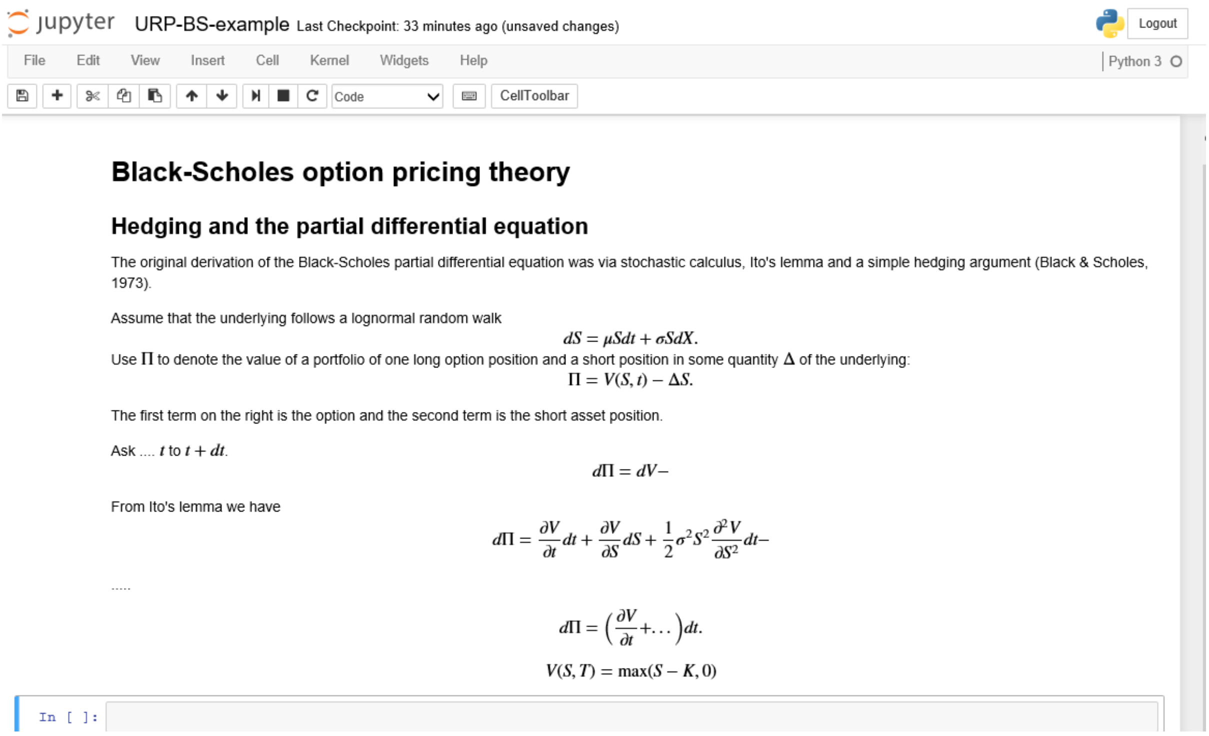


Markdown 으로 변경된 cell 에 URP-BS-example.txt 의 내용을 붙여 넣습니다. Txt 파일은 https://github.com/aorc-group2/URP-FinanceProject 에서 찾을 수 있습니다.




```
# Black-Scholes option pricing theory
## Hedging and the partial differential equation

The original derivation of the Black-Scholes partial differential equation was via stochastic calculus, Ito's lemma and
a simple hedging argument (Black & Scholes, 1973).

Assume that the underlying follows a lognormal random walk
$$
    dS = \mu S dt + \sigma S dX.
$$
Use $\Pi$ to denote the value of a portfolio of one long option position and a short position in some quantity
$\Delta$ of the underlying:
$$
    \Pi = V(S,t) - \Delta S.
$$

The first term on the right is the option and the second term is the short asset position.

Ask .... $t$ to $t+dt$.
$$
    d\Pi = dV -
$$

From Ito's lemma we have
$$
    d\Pi = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS +
    \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt -
$$

.....

$$
    d\Pi = \Big( \frac{\partial V}{\partial t} + ...\Big)dt.
$$

$$
    V(S,T)=\max(S-K,0)
```

이제 cell 안에서 shift + enter 키를 누르면 latex 형식이 수식으로 바뀝니다.




# Black-Scholes option pricing theory

## Hedging and the partial differential equation

The original derivation of the Black-Scholes partial differential equation was via stochastic calculus, Ito's lemma and a simple hedging argument (Black & Scholes, 1973).

Assume that the underlying follows a lognormal random walk

$$dS = \mu S dt + \sigma S dX.$$

Use $\Pi$ to denote the value of a portfolio of one long option position and a short position in some quantity $\Delta$ of the underlying:

$$\Pi = V(S,t) - \Delta S.$$

The first term on the right is the option and the second term is the short asset position.

Ask .... $t$ to $t+dt$.

$$d\Pi = dV -$$

From Ito's lemma we have

$$d\Pi = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt -$$

.....

$$d\Pi = \Big( \frac{\partial V}{\partial t} + ...\Big)dt.$$

$$V(S,T)=\max(S-K,0)$$

앞에서 markdown cell 에서 shift + enter 로 입력한 내용을 다시 편집하고 싶으면 cell 을 더블클릭하면 editor 모드로 전환됩니다.

이제 BSformula_Haug.py 를 두번째 줄 code cell  에 붙여 넣을 차례입니다.

저는 sublime text 라는 text editor 를 사용합니다. 코팅할때 많이 사용하는 편리한 editor 로 쉽게 다운받을 수 있습니다.

BSformula_Haug.py × URP-BS-example.txt ×

```
from math import *
# Cumulative normal distribution

def CND(X):

    (a1,a2,a3,a4,a5) = (0.31938153, -0.356563782, 1.781477937,

     -1.821255978, 1.330274429)
    L = abs(X)

    K = 1.0 / (1.0 + 0.2316419 * L)

    w = 1.0 - 1.0 / sqrt(2*pi)*exp(-L*L/2.) * (a1*K + a2*K*K + a3*pow(K,3) +

    a4*pow(K,4) + a5*pow(K,5))
    if X<0:

        w = 1.0-w

    return w
# Black Sholes Function

def BlackSholes(CallPutFlag,S,X,T,r,v):

    d1 = (log(S/X)+(r+v*v/2.)*T)/(v*sqrt(T))

    d2 = d1-v*sqrt(T)
    if CallPutFlag=='c':

        return S*CND(d1)-X*exp(-r*T)*CND(d2)

    else:

        return X*exp(-r*T)*CND(-d2)-S*CND(-d1)
    
call = BlackSholes('c',100,100,1,0.02,0.2)

print (call)

```

39 lines, 696 characters selected; Copied 696 characters Spaces: 4 Python

두번째 cell 은 code 형식으로 남아있습니다. 여기에 py 코드를 붙여 넣었습니다.

jupyter URP-BS-example Last Checkpoint: 35 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Python 3

Code CellToolbar

$$V(S,T) = \max(S - K, 0)$$

```
In [ ]: from math import *
        # Cumulative normal distribution

        def CND(X):

            (a1,a2,a3,a4,a5) = (0.31938153, -0.356563782, 1.781477937,

             -1.821255978, 1.330274429)
            L = abs(X)

            K = 1.0 / (1.0 + 0.2316419 * L)

            w = 1.0 - 1.0 / sqrt(2*pi)*exp(-L*L/2.) * (a1*K + a2*K*K + a3*pow(K,3) +

            a4*pow(K,4) + a5*pow(K,5))
            if X<0:

                w = 1.0-w

            return w
        # Black Sholes Function

        def BlackSholes(CallPutFlag,S,X,T,r,v):

            d1 = (log(S/X)+(r+v*v/2.)*T)/(v*sqrt(T))

            d2 = d1-v*sqrt(T)
            if CallPutFlag=='c':

                return S*CND(d1)-X*exp(-r*T)*CND(d2)

            else:

                return X*exp(-r*T)*CND(-d2)-S*CND(-d1)

        call = BlackSholes('c',100,100,1,0.02,0.2)

        print (call)
```

Cell 안에서 shift+enter 를 누르면 실행됩니다.

```
    a4*pow(K,4) + a5*pow(K,5))
    if X<0:

        w = 1.0-w

    return w
# Black Sholes Function

def BlackSholes(CallPutFlag,S,X,T,r,v):

    d1 = (log(S/X)+(r+v*v/2.)*T)/(v*sqrt(T))

    d2 = d1-v*sqrt(T)
    if CallPutFlag=='c':

        return S*CND(d1)-X*exp(-r*T)*CND(d2)

    else:

        return X*exp(-r*T)*CND(-d2)-S*CND(-d1)

call = BlackSholes('c',100,100,1,0.02,0.2)

print (call)

8.916035060662303

In [ ]:
```

결과가 수치로 나왔습니다.

이제 이 결과를 ipynb (ipython notebook)으로 다운받습니다.

로컬에 다운받은 결과를 github 에 올리면 됩니다. (github 생성한 URP-.... Repository 로)

Github 에서 ipynb 파일을 더블 클릭하면 jupyer 에서 보던 대로 볼 수 있습니다.