

option_greeks

November 20, 2017

```
In [ ]: import pandas as pd
import datetime
import pickle

expirydate = ('https://www.hkex.com.hk/eng/prod/' +
              'drprod/hkifo/tradcalend_2.htm')

df_d = pd.read_html(expirydate)

date = []
for i in df_d[0][1][16:40]:
    d = datetime.datetime.strptime(i, '%d-%b-%y')
    date.append(d.strftime('%d/%m/%Y'))

with open('/Users/Lwmformula/Downloads/option_trade/'
          'lasttradingday.pkl.txt', 'wb') as f:
    pickle.dump(date, f)

In [ ]: import math
from fractions import gcd
from fractions import Fraction
import pickle

def find_expdays():
    with open('/Users/Lwmformula/Downloads/option_trade/' +
              'lasttradingday.pkl.txt', 'rb') as f:
        date = pickle.load(f)
    today = datetime.datetime.now().strftime('%d,%m,%Y')
    searchele = today.split(',')[1] + today.split(',')[2]
    lasttrading = [int(i[0:2]) for i in date if searchele in i][0]
    expirydate = float(lasttrading - int(today.split(',')[0]))
    return expirydate

def normd2(z):
    b1 = 0.31938153
    b2 = -0.356563782
    b3 = 1.781477937
```

```

b4 = -1.821255978
b5 = 1.330274429
p = 0.2316419

if z > 6.0: return 1
elif z < -6.0: return 0
x = abs(z)
t = 1.0 / (1.0 + p * x)
area = (1.0 - math.exp(-0.5*x**2)/ math.sqrt(2.0*math.pi) *
        (((b5 * t + b4) * t + b3) * t + b2) * t + b1) * t)
if z < 0.0:
    return 1.0 - area
elif z > 0.0: return area

def normd(z):
    return (math.exp(-0.5*z*z)/ math.sqrt(2.0*math.pi))

def BS(underly,strike,expirydate,rfrate,iv, is_call = True):
    t = expirydate / 365.0
    dtmp = iv * math.sqrt(t)
    d1 = (math.log(underly/strike) + rfrate * t)/ dtmp + 0.5 * dtmp
    d2 = d1 - dtmp

    dtmp_2 = strike * math.exp(-rfrate * t)
    vcall = underly * normd2(d1) - dtmp_2 * normd2(d2)
    if is_call == True:
        return vcall
    else: return dtmp_2 + vcall - underly

def IV(underly,strike,optprice,expirydate,rfrate,is_call = True):
    iv = 0.5
    ivstep = 0.25
    price = BS(underly,strike,expirydate,rfrate,iv, is_call)
    while abs(price - optprice) > 0.0000001:
        price = BS(underly,strike,expirydate,rfrate,iv,is_call)
        if ((price > optprice and ivstep > 0.0)
            or (price < optprice and ivstep < 0.0)):
            ivstep = -ivstep * 0.5
        while -ivstep >= iv:
            ivstep *= 0.5
        if iv < 0.0000000001 or iv > 100000.0:
            return 0
        iv += ivstep
    return iv*100.0

def greek(underly,strike,expirydate,iv,rfrate,is_call=True):
    t = expirydate / 365.0
    tsqrt = math.sqrt(t)

```

```

    dtmp = iv * tsqrt
    d1 = (math.log(underly/strike) + rfrate * t) / dtmp + 0.5 * dtmp
    d2 = d1 - dtmp
    n_d1 = normd(d1)
    n_d2 = normd2(d2)

    if is_call is True:
        delta = normd2(d1)
        theta = (-(underly*iv*n_d1)/(2.0*math.sqrt(t)) -
                 rfrate*strike*math.exp(-rfrate*t)*n_d2)
        rho = 0.01 * strike * t * math.exp(-rfrate*t) * n_d2
    elif is_call is False:
        delta = normd2(d1) - 1.0
        theta = ((-underly*iv*n_d1)/(2.0*math.sqrt(t)) +
                 rfrate*strike*math.exp(-rfrate*t)*(1.0-n_d2))
        rho = -0.01 * strike * t * math.exp(-rfrate * t) * (1.0 - n_d2)

    gamma = n_d1 / (underly*dtmp)
    vega = 0.01 * underly * math.sqrt(t) * n_d1
    theta = theta / 365.0

    return delta,gamma,theta,vega,rho

cunderly = 14.62
cstrike = 14.0
crfrate = 0
#0.27/100
coptprice = 0.86
cexpirydate = 11
#find_expiryday()
call_iv = IV(cunderly,cstrike,coptprice,
             cexpirydate,crfrate,is_call = True)
cdelta,cgamma,ctheta,cvega,crho = greek(cunderly,
                                         cstrike,cexpirydate,
                                         call_iv/100.0,crfrate,is_call=True)

punderly = 14.62
pstrike = 14.0
prfrate = 0
#0.27/100
poptprice = 0.29
pexpirydate = 11
#find_expiryday()
put_iv = IV(punderly,pstrike,poptprice,
            pexpirydate,prfrate,is_call = False)
pdelta,pgamma,ptheta,pvega,prho = greek(punderly,pstrike,
                                         pexpirydate,put_iv/100.0,

```

```

prfrate,is_call=False)

ratio = abs(cdelta)/abs(pdelta)

ratio_opt = []
for i in range(5,51):
    ratio_tmp = []
    ratio_tmp.append(str(Fraction(str(ratio)).limit_denominator(i)).split('/')[0])
    ratio_tmp.append(str(Fraction(str(ratio)).limit_denominator(i)).split('/')[1])
    if ratio_tmp in ratio_opt:
        continue
    else: ratio_opt.append(ratio_tmp)

choice = []
for i in ratio_opt:
    sim_tmp = round(float(i[0])/float(i[1]),5)
    choice.append([i[0],i[1],sim_tmp,round(ratio,5)])

print choice

```